Hyperbolic Deep Learning in Computer Vision: A Survey

Pascal Mettes[1] · Mina Ghadimi Atigh[1] · Martin Keller-Ressel[2] · Jeffrey Gu[3] · Serena Yeung[3]

Abstract

Deep representation learning is a ubiquitous part of modern computer vision. While Euclidean space has been the de facto standard manifold for learning visual representations, hyperbolic space has recently gained rapid traction for learning in computer vision. Specifically, hyperbolic learning has shown a strong potential to embed hierarchical structures, learn from limited samples, quantify uncertainty, add robustness, limit error severity, and more. In this paper, we provide a categorization and in-depth overview of current literature on hyperbolic learning for computer vision. We research both supervised and unsupervised literature and identify three main research themes in each direction. We outline how hyperbolic learning is performed in all themes and discuss the main research problems that benefit from current advances in hyperbolic learning for computer vision. Moreover, we provide a high-level intuition behind hyperbolic geometry and outline open research questions to further advance research in this direction.

Keywords Hyperbolic deep learning · Computer vision · Representation learning

1 Introduction

From image segmentation to future frame prediction and from video grounding to generating images, deep representation learning is the central component that drives modern computer vision problems (LeCun et al., 2015). In short succession, many differentiable layers and network architectures have been proposed to tackle visual research problems (Gu et al., 2018; Bommasani et al., 2021; Khan et al., 2022). While different in structure, scope, and inductive biases, all are based on Euclidean operators and therefore - implicitly or explicitly - assume that data is best represented on regular grids.

Euclidean space forms an intuitive and grounded underlying manifold, but its inherent properties are not a best match for all types of data. Consider for example hierarchical structures such as trees, ontologies, and taxonomies. Hierarchies are foundational building blocks across all scientific disciplines to formalize our knowledge (Noy & Hafner, 1997). In hierarchies, the number of nodes grows exponentially with depth, from few coarse-grained to many fine-grained nodes. The volume of a ball in Euclidean space however, grows only polynomially with its diameter. An alternative geometry is needed to match the nature of hierarchies.

In the quest for a more appropriate geometry of hierarchies, hyperbolic geometry provides a direct fit (Bridson & Haefliger, 2013). In essence, hyperbolic and Euclidean geometry are different in only one aspect: the parallel postulate. In Euclidean space, there is exactly one parallel line that goes through a point not on the other line. In hyperbolic space, there are at least two such parallel lines. This change comes with many consequences and as a result, hyperbolic geometry can be seen as a geometry of constant negative curvature. In the context of deep learning this geometry has many attractive properties, such as its hierarchical structure and exponential expansion.

✉ Pascal Mettes
p.s.m.mettes@uva.nl

Mina Ghadimi Atigh
m.ghadimiatigh@uva.nl

Martin Keller-Ressel
martin.keller-ressel@tu-dresden.de

Jeffrey Gu
jeffgu@stanford.edu

Serena Yeung
syyeung@stanford.edu

[1] University of Amsterdam, Amsterdam, The Netherlands

[2] TU Dresden, Dresden, Germany

[3] Stanford University, Stanford, CA, USA

Empowered by these geometric properties, hierarchical embeddings have in recent years been performed in hyperbolic space with great success (Nickel & Kiela, 2017), leading to unparalleled abilities to embed deep and complex trees with minimal distortion (Ganea et al., 2018a; Sala et al., 2018; Sonthalia & Gilbert, 2020; Verbeek & Suri, 2014; Chami et al., 2020a). This has led to rapid advances in hyperbolic deep learning across many disciplines and research areas, including but not limited to graph networks (Chami et al., 2019; Liu et al., 2019; Dai et al., 2021; Sarkar, 2011; Sun et al., 2021a; Yang et al., 2022a; Wang et al., 2023b), text embeddings (Tifrea et al., 2019; Zhu et al., 2020; Dhingra et al., 2018; Dai et al., 2020), molecular representation learning (Klimovskaia et al., 2020; Yu et al., 2020; Wu et al., 2021; Qu & Zou, 2022b), and recommender systems (Mirvakhabova et al., 2020; Wang et al., 2021; Yang et al., 2022b; Li et al., 2022; Vinh Tran et al., 2020; Chamberlain et al., 2019; Vinh et al., 2018).

In the wake of other disciplines, computer vision has in recent years also benefited from research into deep learning in hyperbolic space. A quickly growing body of literature has shown that hyperbolic embeddings benefit few-shot learning (Fang et al., 2021; Khrulkov et al., 2020; Gao et al., 2021; Guo et al., 2022), zero-shot recognition (Long et al., 2020; Liu et al., 2020; Ghadimi Atigh et al., 2021; Hong et al., 2023b), out-of-distribution generalization (Khrulkov et al., 2020; Hong et al., 2023a; Guo et al., 2022), uncertainty quantification (Khrulkov et al., 2020; Ghadimi Atigh et al., 2022; Chen et al., 2022), generative learning (Kingma & Welling, 2013; Rezende et al., 2014; Lazcano et al., 2021; Heusel et al., 2017), and hierarchical representation learning (Dhall et al., 2020; Long et al., 2020; Gulshad et al., 2023; Liu et al., 2020; Ghadimi Atigh et al., 2022) amongst others. These works show evidence that hyperbolic geometry has a lot of potential for learning in computer vision.

This survey provides an in-depth overview and categorization of the recent boom in hyperbolic computer vision literature. These works have investigated hyperbolic learning across many visual research problems with different solutions. As a result, it is unclear how current literature is connected, what is common and new in each work, and in which direction the field is heading. This survey seeks to fill this void. We investigate both supervised and unsupervised papers. For supervised learning, we identify three shared themes amongst current papers, where samples are matched to either gyroplanes, prototypes, or other samples in hyperbolic space. For unsupervised papers, we dive into the three main axes explored in current papers, namely generative learning, clustering, and self-supervised learning. This survey fills this void. Peng et al. (2021) have recently written a general survey on hyperbolic neural networks but their main focus is not on advances in computer vision. This survey fills this void. Fang et al. (2023b) have made a con-

current overview of hyperbolic learning in the context of computer vision. Our survey extends the survey of Fang et al. (2023b) by providing a grouping of the advances in supervised and unsupervised hyperbolic learning, delivering an in-depth overview of hyperbolic geometry with its most important functionalities for deep learning, and discussing emerging advances such as fully hyperbolic learning.

The rest of the paper is organised as follows. In Sect. 2 we provide the background on hyperbolic geometry and foundational papers on hyperbolic embeddings and hyperbolic neural networks. Sections 3 and 4 provide an overview of supervised and unsupervised hyperbolic visual learning literature. Lastly in Sect. 5 we outline advantages and improvements reported in current papers, as well as open challenges for the field.

2 Background on Hyperbolic Geometry

2.1 What is Hyperbolic Geometry?

Hyperbolic geometry was initially developed in the 19th century by Gauss, Lobachevsky, Bolyai and others as a concrete example of a non-Euclidean geometry (Cannon et al., 1997). Soon after it found important applications in physics, as the mathematical basis of Einstein's special theory of relativity. It can be characterized as the geometry of *constant negative curvature*, differentiating it from the flat geometry of Euclidean space and the positively curved geometry of spheres and hyperspheres. From the point of view of representation learning, its attractive properties are its exponential expansion and its hierarchical, tree-like structure. Exponential expansion means that the volume of a ball in hyperbolic space growths exponentially with its diameter, in contrast to Euclidean space, where the rate of growth is polynomial. The 'tree-likeness' of a metric space can be quantified by Gromov's hyperbolicity (Bridson & Haefliger, 2013), which is zero for tree graphs, finite (but non-zero) for hyperbolic space, and infinite for Euclidean space.

2.2 Models of Hyperbolic Geometry

Several different, but isometric, models of hyperbolic geometry exist (Cannon et al., 1997). They differ in their coordinate representations of points and in their expressions for distances, geodesics, and other quantities. Although they can be isometrically mapped to each other, certain models may be preferred for a given task, for reasons of numerical efficiency, ease of visualization, or simplified calculations. The most commonly used models are the Poincaré model, the hyperboloid (or 'Lorentz') model, the Klein model, and the upper half-space model.

Fig. 1 Circle Limit I (1958). This artwork by M. C. Escher is based on the Poincaré disc model of hyperbolic geometry

Fig. 2 Hyperboloid and Poincaré disc model. This figure shows the relationship between the hyperboloid model and the Poincaré model of hyperbolic geometry. In each model, two points (red) and their connecting geodesic arc (blue) are shown, as well as the tangent plane (light blue) at one of the points in the hyperboloid model

- The **Poincaré model** represents d-dimensional hyperbolic space by the unit ball
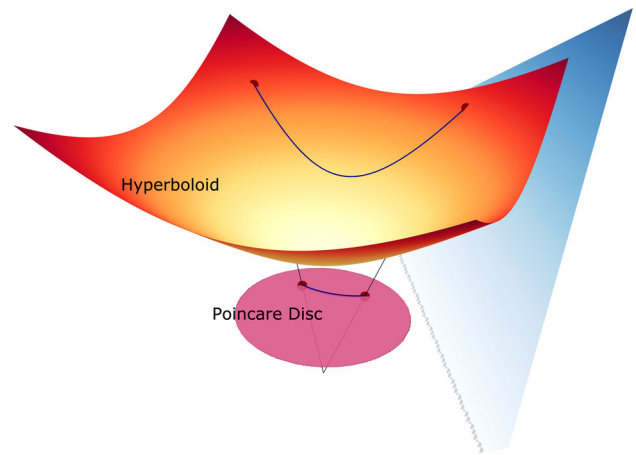
$$\mathbb{D}_d = \{p \in \mathbb{R}^d : p_1^2 + \cdots + p_d^2 < 1\}$$

which, in the frequently considered case $d = 2$ becomes the unit disc. Geodesics ('shortest paths') are arcs of Euclidean circles (or lines), meeting the boundary of $\mathbb{D}_d$ at a right angle. While distances, area and volume are distorted in comparison to their Euclidean counterparts, the model is *conformal*, i.e., hyperbolic angles are measured as in Euclidean geometry. In its two-dimensional form as Poincaré disc, the model is popular for visualizations; it is also the geometric basis of the art works Circle Limits I-IV of M. C. Escher; see Fig. 1.

- The **hyperboloid model** uses the single-sheet hyperboloid

$$\mathbb{H}_d = \{p \in \mathbb{R}^{d+1} : p_0^2 - \left(p_1^2 + \cdots + p_d^2\right) = 1, \, p_0 > 0\}$$

as a model of d-dimensional hyperbolic geometry. Contrary to the other models, its ambient space $\mathbb{R}^{d+1}$ adds one dimension to the modeled space. Many formulas involving the hyperboloid model can be written in concise form by introducing the *Lorentz product* $p \circ q = p_0 q_0 - (p_1 q_1 + \cdots + p_d q_d)$. An advantage of the hyperboloid model is that it retains some linear structure; translations and other isometries, for example, can be

represented by linear maps. Expressions for distances and geodesics are simpler compared to other models. Notably, the Poincaré model can be derived as a projection ('stereographic projection') of the hyperboloid model to the unit ball (Cannon et al., 1997; Ratcliffe, 1994). Fig. 2 shows how the hyperboloid model and the Poincaré ball model are related.

- The **Klein model** $\mathbb{K}_d$ also uses the unit ball to represent hyperbolic space. In contrast to the Poincaré model, it is not conformal; its geodesics, however, are Euclidean ('straight') lines, which can be beneficial from a computational point of view, e.g., when computing barycenters.

- Lastly, the **upper half space model** represents d-dimensional hyperbolic space by the set $\mathbb{U}_d = \{p \in \mathbb{R}^d : p_d > 0\}$. It is a conformal model and shares many properties with the Poincaré model; geodesics, for example, are also arcs of Euclidean circles (or lines), meeting the boundary of $\mathbb{U}_d$ at a right angle.

2.3 Five core Hyperbolic Operations

Within the context of deep learning and computer vision, we find that five core operations form the basic building blocks of the vast majority of algorithms that use hyperbolic geometry for learning. The ability to work with these five operations will cover most of the existing literature:

1. Measuring the **distance** of two points p and q;
2. Finding the **geodesic arc** (the distance-minimizing curve) from p to q;
3. Forming a **geodesic**, by extending a geodesic arc as far as possible;

4. Using the **exponential map**, to determine the result of following a geodesic in direction u, at speed r, starting at a point p;

5. Moving a cloud of points, while preserving all their pairwise hyperbolic distances, by applying a **hyperbolic translation**.

The distance of two points is given, in the Poincaré and the hyperboloid model respectively, by

$$d_{\mathbb{D}}(p, q) = \frac{1}{\sqrt{\kappa}} \operatorname{arcosh}\left(1 + \frac{2|p-q|^2}{(1-|p|^2)(1-|q|^2)}\right), \quad (1)$$

$$d_{\mathbb{H}}(p, q) = \frac{1}{\sqrt{\kappa}} \operatorname{arcosh}(p \circ q). \quad (2)$$

In the less frequently used Klein and the upper half space model, distances are given by

$$d_{\mathbb{K}}(p, q) = \frac{1}{\sqrt{\kappa}} \operatorname{arcosh}\left(\frac{1 - p^\top q}{\sqrt{1-|p|^2}\sqrt{1-|q|^2}}\right), \quad (3)$$

$$d_{\mathbb{U}}(p, q) = \frac{1}{\sqrt{\kappa}} \operatorname{arcosh}\left(1 + \frac{|p-q|^2}{2p_d q_d}\right), \quad (4)$$

see Ratcliffe (1994, §6.1). The scaling factor of distances is controlled by the *curvature parameter* $\kappa \in (0, \infty)$, which is often standardized to $\kappa = 1$. The sectional curvature (in the sense of differential geometry) of hyperbolic space is constant, negative and equal to $-\kappa$. Given the distance function, it makes sense to speak of geodesics and geodesic arcs, that is (locally) distance-minimizing curves, either extending infinitely or connecting two points. In the hyperboloid model for example, each geodesic is the intersection of $\mathbb{H}_d$ with a Euclidean hyperplane in the ambient space $\mathbb{R}^{d+1}$. The geodesic at a point $p \in \mathbb{H}_d$ in direction u can be written as

$$\lambda_{\mathbb{H}}(t) = \cosh(t\sqrt{\kappa})p + \sinh(t\sqrt{\kappa})u, \quad t \in \mathbb{R}. \quad (5)$$

where u is an element of the *tangent space* $T_p = \{u \in \mathbb{R}^{d+1} : p \circ u = 0\}$, normalized to $u \circ u = -1$. In the Poincaré model, the geodesics are precisely the segments of Euclidean circles and lines that meet the boundary of $\mathbb{D}_d$ at a right angle. A convenient formula for the geodesic arc between two points $p, q \in \mathbb{D}_d$ can be given in terms of gyrovectorspace calculus, see (8).

The value of the exponential map $\exp_p(tu)$ is the result of following a geodesic in a normalized direction u at a speed $t > 0$, after starting at a given point p in hyperbolic space. Identifying $\mathbb{R}^d$ with the tangent space T_p at p, the exponential mapping provides a convenient way to embed $\mathbb{R}^d$ into hyperbolic space with origin at p. The exponential map is the most often used function in hyperbolic learning for computer vision, as it allows us to map visual representations from Euclidean to hyperbolic space. In the hyperboloid model, the exponential mapping coincides with the expression of the geodesic given in (5). In the Poincaré model the exponential map can be conveniently written in terms of gyrovectorspace addition and is given in (9). In practice, the exponential and logarithmic mapping functions are tools in vision for mapping representations from Euclidean to hyperbolic space or vice versa. This is common for example when using hyperbolic embeddings on top of standard encoders or when using pre-trained networks.

Finally, the hyperbolic translation τ_p, also called Lorentz boost, Möbius transformation, or gyrovectorspace addition, is the unique distance-preserving transformation of hyperbolic space, which moves 0 to a given point p. Concatenations of logarithmic maps, parallel transport in the tangent space and exponential maps, as used for example in Ganea et al. (2018a) can be expressed in terms of hyperbolic translations, or equivalently in terms of gyrovectorspace addition; see Eq. (26) in Ganea et al. (2018a). In the hyperboloid model, the hyperbolic translation can be represented by the linear map

$$\tau_p(q) = L_p \cdot q, \quad \text{where} \quad (6)$$

$$L_p = \begin{pmatrix} p_0 & \bar{p}^\top \\ \bar{p} & \sqrt{I_d + \bar{p}\bar{p}^\top} \end{pmatrix} \text{ with } \bar{p} = (p_0, \dots, p_d). \quad (7)$$

In the Poincaré model hyperbolic translations are also known as gyrovectorspace addition and form the basic operation of gyrovectorspace calculus. For the equivalence of gyrovectorspace addition and hyperbolic translations, one can compare Eq.(4) in Ganea et al. (2018a) and Eq. (4.5.5) in Ratcliffe (1994). For the equivalence of hyperbolic translations and Lorenz boosts see e.g., Sec. 2.2. in Chen et al. (2021).

2.4 Gyrovectorspace Calculus

Gyrovectorspace calculus, as introduced by Ungar (2005, 2012), provides a convenient and rapidly adopted framework for calculations in the Poincaré ball model. Its first basic operation is the (non-commutative) gyrovectorspace addition

$$p \oplus q = \frac{(1 - |p|^2)q + (1 + 2p^\top q + |q|^2)p}{1 + 2p^\top q + |p|^2|q|^2}.$$

As a secondary operation, the (commutative) gyrovectorspace scalar product

$$t \otimes p = p \otimes t = \tanh\left(t \operatorname{artanh}(|p|)\right)\frac{p}{|p|}$$

with a scalar $t \in \mathbb{R}$ is introduced. Hyperbolic translations are directly given by $\tau_p(q) = p \oplus q$ and the geodesic arc

connecting p and q is

$$\lambda_{\mathbb{D}}(t) = p \oplus \left(\left((-p) \oplus q \right) \otimes t \right), \quad t \in [0, 1]. \tag{8}$$

Letting t range through all of $\mathbb{R}$ a full geodesic line is obtained.

In the context of gyrovector space calculus, the Poincaré ball is often rescaled with the square root of curvature, setting

$$\mathbb{D}_\kappa^d = \{p \in \mathbb{R}^d : p_1^2 + \cdots + p_d^2 < 1/\kappa\}.$$

The advantage of this rescaling is that Euclidean space is obtained as a continuous limit as $\kappa \to 0$. In the rescaled model, gyrovectorspace addition and scalar product become

$$p \oplus_\kappa q = \tfrac{1}{\sqrt{\kappa}} \left((\sqrt{\kappa} p) \oplus (\sqrt{\kappa} q) \right)$$

and

$$t \otimes_\kappa p = \tfrac{1}{\sqrt{\kappa}} (t \otimes (\sqrt{\kappa} p))$$

for $p, q \in \mathbb{D}_\kappa^d$. The exponential map in the direction of a tangent vector $v \in T_p$ can then be written as

$$\exp_p^\kappa(v) = x \oplus_\kappa \left(\tanh \left(\frac{\sqrt{\kappa}|v|}{1 - \kappa|p|^2} \right) \frac{v}{\sqrt{\kappa}|v|} \right) \tag{9}$$

for $p \in \mathbb{D}_\kappa^d$, see Ganea et al. (2018b).

2.5 Non-visual Hyperbolic Learning

The traction of hyperbolic learning in computer vision is built upon advances in embedding hierarchical structures, designing hyperbolic network layers, and hyperbolic learning on other data types such as graphs, text, and more. Below, we discuss these works and their relevance for hyperbolic visual learning literature.

Hyperbolic embedding of hierarchies. Embedding hierarchical structures like trees and taxonomies in Euclidean space suffers from large distortion (Bachmann et al., 2020), and polynomial volume expansion, limiting its capacity to capture the exponential complexity of hierarchies. However, hyperbolic space can be thought of as a continuous version of trees (Nickel & Kiela, 2017) and has tree-like properties (Hamann, 2018; Ungar, 2008), like the exponential growth of distances when moving from the origin towards the boundary. Encouraged by this, Nickel & Kiela (2017) propose to embed hierarchical structures on the Poincaré model. The goal is to learn hyperbolic representations for the nodes of a hierarchy, such that the distance in the embedding space has an inverse relation with semantic similarity. Let $\mathcal{D} = \{(u, v)\}$ denote the set of the nodes connected in a given hierarchy. To embed the nodes in the Poincaré model, Nickel & Kiela (2017) minimize the following loss function:

$$\mathcal{L}(\Theta) = \sum_{(u,v) \in \mathcal{D}} \log \frac{e^{-d(u,v)}}{\sum_{v' \in \mathcal{N}(u)} e^{-d(u,v')}}, \tag{10}$$

where $\mathcal{N}(u) = \{v' | (u, v') \notin \mathcal{D}\} \cup \{v\}$ denotes the set of the nodes not related to u, including v, as negative examples. The loss function pushes unrelated nodes farther apart than the related ones. To evaluate the embedded hierarchy, the distances between pairs of connected nodes (u, v) are calculated and ranked among the negative pairs of nodes (i.e., the nodes not in $\mathcal{D}$), and the mean average precision (MAP) is calculated based on the ranking. Later, Sala et al. (2018) propose a combinatorial construction to embed the trees in hyperbolic space without optimization and with low distortion, relieving the optimization problems in existing works. Ganea et al. (2018a) address drawbacks of Nickel & Kiela (2017) including the collapse of the points on the boundary of the space as a result of the loss function and incapability of encoding asymmetric relations. They introduce entailment cones to embed hierarchies, using a max-margin loss function:

$$\mathcal{L} = \sum_{(u,v) \in \mathcal{P}} E(u, v) + \sum_{(u',v') \in \mathcal{N}} \max(0, \gamma - E(u', v')), \tag{11}$$

where γ, $\mathcal{P}$, and $\mathcal{N}$ indicate margin, the positive and negative edges, respectively. $E(u, v)$ is a penalty term that forces child nodes to fall under the cone of the parent node. Amongst others, hyperbolic embeddings have been proposed for multi-relational graphs (Balazevic et al., 2019), low-dimensional knowledge graphs (Chami et al., 2020b), and learning continuous hierarchies in Lorentz model given pairwise similarity measurements (Nickel & Kiela, 2018). Nickel & Kiela (2018) proposes to learn embeddings $\Theta = \{u\}_{i=1}^m$ in the Lorentz model by optimizing

$$\max_\Theta \sum_{i,j} \log Pr(\phi(i, j) = j | \Theta) \tag{12}$$

where given $\mathcal{N}(i, j)$ as the set concepts to embed,

$$\phi(i, j) = \underset{z \in \mathcal{N}(i,j)}{\arg \min} d(u_i, u_z)$$

$$Pr(\phi(i, j) = j | \Theta) = \frac{e^{-d(u_i, u_j)}}{\Sigma_{z \in \mathcal{N}(i,j)} e^{-d(u_i, u_z)}}.$$

Hyperbolic neural networks. Foundational in the transition of deep learning towards hyperbolic space is the development of hyperbolic network layers and their optimization.

We consider two pivotal papers here that provide a such theoretical foundation, namely Hyperbolic Neural Networks by Ganea et al. (2018b) and Hyperbolic Neural Networks++ by Shimizu et al. (2021). Ganea et al. (2018b) propose multinomial logistic regression in the Poincaré ball.

Given $k \in \{1, ..., K\}$ classes, $p_k \in \mathbb{D}_c^n$, $\forall q \in \mathbb{D}_c^n$, and $a_k \in \mathbb{D}_c^n \backslash \{0\}$, hyperbolic logistic regression is performed using

$$p(y = k|q) \propto \exp\left(\frac{\lambda_{p_k}^c \|a_k\|}{\sqrt{c}} \right.$$
$$\left. \sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_k \oplus_c q, a_k\rangle}{(1 - c\|-p_k \oplus_c q\|^2)\|a_k\|}\right)\right). \quad (13)$$

Intuitively, the above equation describes the distance to the margin hyperplane in hyperbolic space. As an extension, a hyperbolic version of linear layer f is given as $f : \mathbb{R}^n \to \mathbb{R}^m$, a Möbius version of f where the map from $\mathbb{D}^n \to \mathbb{D}^m$ is defined as:

$$f^{\otimes_c} := \exp_0^c(f(\log_0^c(q))), \quad (14)$$

with $\exp_0^c : T_{0_m}\mathbb{D}_c^m \to \mathbb{D}_c^m$ and $\log_0^c : \mathbb{D}_c^n \to T_{0_n}\mathbb{D}_c^n$. They furthermore outline how to create recurrent network layers. Bdeir et al. (2023) also provide the Lorentzian formulation of 2D convolutional layer, batch normalization, and multinomial logistic regression. As (Bdeir et al., 2023) show, given parameters $a_c \in \mathbb{R}$ and $z_c \in \mathbb{R}^n$, the logit for class c and input $x \in \mathbb{L}_K^n$ is given as:

$$v_{z_c, a_c}(x) = \frac{1}{\sqrt{-K}} sign(\alpha)\beta|\sinh^{-1}(\sqrt{-K}\frac{\alpha}{\beta})|$$
$$\alpha = \cosh(\sqrt{-K}a)\langle z, x_s\rangle - \sinh(\sqrt{-K}a) \quad (15)$$
$$\beta = \sqrt{\|\cosh(\sqrt{-K}a)z\|^2 - (\sinh(\sqrt{-K}a)\|z\|)^2}.$$

Shimizu et al. (2021) reformulate the hyperbolic logistic regression of Ganea et al. (2018b) to reduce the number of parameters to the same level as the Euclidean logistic regression. Their linear layer is given as:

$$y = \mathcal{F}^c(p; Z, r) := w(1 + \sqrt{1 + c\|w\|^2})^{-1} \quad (16)$$

where $Z = \{z_k \in T_0\mathbb{B}_c^n = \mathbb{R}^n\}_{k=1}^m$, $r = \{r_k \in \mathbb{R}\}_{k=1}^m$, and $w := (c^{-\frac{1}{2}}\sinh(\sqrt{c}v_k(p)))_{k=1}^m$. More importantly for computer vision, they show how to formulate convolutional layers using Poincaré fully connected layer and β-concatenation. To do so, they show how to generalize the hyperbolic linear layer to image patches through β-splits, and β-concatenation, leading in principle to arbitrary-dimensional convolutional layers. Moreover, Poincaré multi-head attention is possible through the same operators.

Following Ganea et al. (2018b) and Shimizu et al. (2021), Yang et al. (2023) investigate the hierarchical representation ability of the existing HNNs and HGNNs, improving the hierarchical ability through hyperbolic informed embedding (HIE) via incorporating hierarchical distance of the node to origin. HIE is task- and model-agnostic and can be used to improve the hierarchical embedding ability of different hyperbolic models (i.e., Poincaré model and Lorentz model). Park et al. (2023) use hyperbolic neural networks and propose a Hyperbolic Affinity Learning method for spatial propagation and learning the hierarchical relationship among the pixels.

Hyperbolic learning of graphs, text, and more. The advances in hyperbolic embeddings of hierarchies and the introduction of hyperbolic network layers have spurred research in several other research directions as well. As a logical extension of hierarchical embeddings, graph networks have been extended to hyperbolic space. Liu et al. (2019) and Chami et al. (2019) propose a tangent-based view to hyperbolic graph networks. Both approaches model a graph layer by first mapping node embeddings to the tangent space, then performing the transformation and aggregation in the tangent space, after which the updated node embeddings are projected back to the hyperbolic manifold at hand. Since tangent operations only provide an approximation of the graph operations on the manifold, several works have proposed graph networks that better abide the underlying hyperbolic geometry, such as constant curvature κ-GCNs (Bachmann et al., 2020), hyperbolic-to-hyperbolic GCNs (Dai et al., 2021), Lorentzian GCNs (Zhang et al., 2021c), Lorentzian nested hyperbolic GCNs (Fan et al. 2022), attention-based hyperbolic graph networks (Gulcehre et al., 2019; Zhang et al., 2021b), dynamic hyperbolic graph attention network (Li et al., 2023a), and embedding graphs by combining hyperbolic and diffusion geometry (Lin et al., 2023c). Hyperbolic graph networks have shown to improve node, link, and graph classification compared to Euclidean variants, especially when graphs have latent hierarchical structures.

Hyperbolic embeddings have also been investigated for text. Tifrea et al. (2019), Dhingra et al. (2018), and Leimeister & Wilson (2018) propose hyperbolic alternatives for word embeddings. Zhu et al. (2020) introduce HyperText to endow FastText with hyperbolic geometry. Embedding text in hyperbolic space has the potential to improve similarity, analogy, and hypernymy detection, most notably with few embedding dimensions.

Beyond text and graphs, hyperbolic learning has shown to be beneficial for several other research directions, including but not limited to learning representations for molecular/cellular structures (Klimovskaia et al., 2020; Yu et al., 2020; Wu et al., 2021), recommender systems (Mirvakhabova et al., 2020; Wang et al., 2021; Yang et al., 2022b), reinforcement learning (Cetin et al., 2022), music genera-
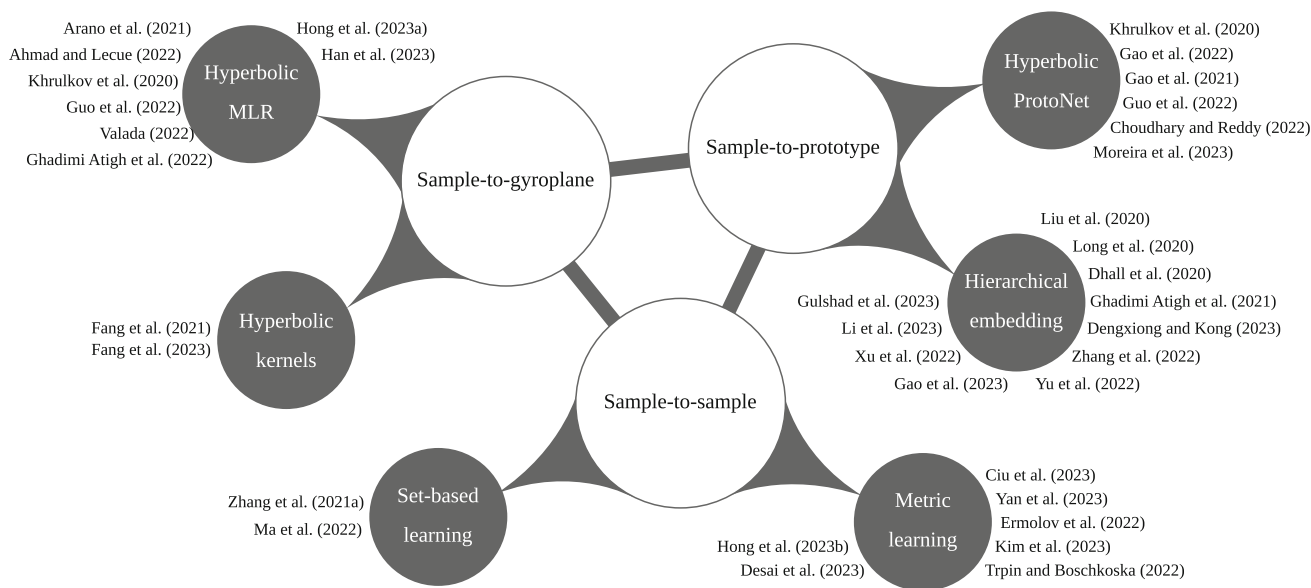
Fig. 3 **The three core strategies for supervised hyperbolic learning in computer vision.** Current literature performs hyperbolic learning of visual embeddings by learning to match training samples (i) to hyperbolic class hyperplanes, i.e., gyroplanes, (ii) to hyperbolic class prototypes, or (iii) by contrasting to other samples

tion (Huang et al., 2023), skeletal data (Franco et al., 2023; Chen et al., 2023), LiDAR data (Tong et al., 2022; Wang et al., 2023a), point clouds (Montanaro et al., 2022; Anvekar & Bazazian, 2023; Lin et al., 2023b; Onghena et al., 2023), 3D shapes (Chen et al., 2020b; Onghena et al., 2023; Leng et al., 2023), and remote sensing data (Hamzaoui et al., 2023). In summary, hyperbolic geometry has impacted a wide range of research fields. This survey focuses specifically on the impact and potential in the visual domain.

3 Supervised Hyperbolic Visual Learning

In Fig. 3, we provide an overview of literature on supervised learning with hyperbolic geometry in computer vision. In current vision works, hyperbolic learning is mostly performed at the embedding- or classifier-level. In other words, current works rely on standard networks for feature learning and transform the output embeddings to hyperbolic space for the final learning stage. For supervised learning in hyperbolic space, we have identified three main optimization strategies:

1. *Sample-to-gyroplane learning* denotes the setting where classes are represented by hyperbolic hyperplanes, i.e., gyroplanes, with networks optimized based on confidence logit scores between samples and gyroplanes.
2. *Sample-to-prototype learning* denotes the setting where class semantics are represented as points in hyperbolic

space, and networks are optimized to minimize hyperbolic distances between samples and prototypes.
3. *Sample-to-sample learning* denotes the setting where networks are optimized by learning metrics or contrastive objectives between samples in a batch.

For all strategies, let (x, y) denote the visual input x, which can be an image or a video, and the corresponding label $y \in \mathcal{Y}$. Let $f_\theta(x) \in \mathbb{R}^D$ denote its Euclidean embedding after going through a network. This representation is mapped to hyperbolic space using the exponential map, denoted as $g(x) = \exp_0(f_\theta(x))$. In many hyperbolic works, additional information about hierarchical relations between classes is assumed. Let $\mathcal{H} = (\mathcal{Y}, \mathcal{P}, \mathcal{R})$, with $\mathcal{Y}$ the class labels denoting the leaf nodes of the hierarchy, $\mathcal{P}$ the internal nodes, and $\mathcal{R}$ the set of hypernym-hyponym relations of the hierarchy. Below, we discuss how current literature tackles each strategy in detail sequentially.

3.1 Sample-to-Gyroplane Learning

The most direct way to induce hyperbolic geometry in the classification space is by replacing the classification layer by a hyperbolic alternative. This can be done either by means of a hyperbolic logistic regression or through hyperbolic kernel machines.
Hyperbolic logistic regression. Khrulkov et al. (2020) incorporate a hyperbolic classifier by taking a standard convolutional network and mapping the outputs of the last hidden

layer to hyperbolic space using an exponential map. Afterwards, the hyperbolic multinomial logistic regression as described by Ganea et al. (2018b) is used to obtain class logits which can be optimized with cross-entropy. They find that training a hyperbolic classifier on top of a convolutional network allows us to obtain uncertainty information based on the distance to the origin of the hyperbolic embeddings of images. Out-of-distribution samples on average have a smaller norm, making it possible by differentiating in- to out-of-distribution samples by sorting them by the distance to the origin. Hong et al. (2023a) show that hyperbolic classification is beneficial for visual anomaly recognition tasks, such as out-of-distribution detection in image classification and segmentation tasks. Araño et al. (2021) use hyperbolic layers to perform multi-modal sentiment analysis based on the audio, video, and text modalities. Ahmad & Lecue (2022) also show the effect of hyperbolic space to perform object recognition with ultra-wide field-of-view lenses. Han et al. (2023) show that hyperbolic embeddings with logistic regression and an extra contrastive loss benefits anti-face spoofing.

Guo et al. (2022) address a limitation when training classifiers in hyperbolic space, namely a vanishing gradient problem due to the hybrid architecture of current hyperbolic approaches in computer vision, where Euclidean features are connected to a hyperbolic classifier. Equation 13 highlights that to maximize the likelihood of correct predictions, the distance to hyperbolic gyroplanes needs to be maximized. In practice, embeddings of samples are pushed to the boundary of the Poincaré ball. As a result, the inverse of the Riemannian tensor metric approaches zero, resulting in small gradients. This finding is in line with several other works on vanishing gradients in hyperbolic representation learning in Poincaré and Lorentz models (Nickel & Kiela, 2018; Liu et al., 2019).

To combat the vanishing gradient problem, Guo et al. (2022) propose to clip the Euclidean embeddings of samples before the exponential mapping, i.e.:

$$f_\theta^{\text{clipped}}(x) = \min\left\{1, \frac{r}{||f_\theta(x)||}\right\} \cdot f_\theta(x), \qquad (17)$$

with r as a hyperparameter. This trick improves learning with hyperbolic multinomial logistic regression, especially when dealing with many classes such as on ImageNet. Furthermore, training with clipped hyperbolic classifiers improves out-of-distribution detection over training with Euclidean classifiers, while also being more robust to adversarial attacks. However, Moreira et al. (2023) dive into the hyperbolic prototypical networks with high-dimensional output space while performing a few-shot learning task, showing that the hyperbolic representations concentrate close to the surface, resulting in a boundary saturation. Mishne et al. (2023) analyze the limitations and differences between Poincaré and Lorentz models, along with a Euclidean

parametrization to hyperbolic space. These works indicate the need for more robust representation and optimization when working in hyperbolic space.

Next to global classification, a few recent works have investigated hyperbolic logistic regression for structured prediction tasks such as object detection and image segmentation. Valada (2022) extend object detection with hyperbolic geometry, amongst others by replacing the classifier head of a two-stage detection like Sparse R-CNN (Sun et al., 2021b) with a hyperbolic logistic regression, improving object detection performance in standard and zero-shot settings. Ghadimi Atigh et al. (2022) introduce Hyperbolic Image Segmentation, where the final per-pixel classification was performed in hyperbolic space. Starting from the geometric interpretation of hyperbolic gyroplanes of Ganea et al. (2018b), they find that simultaneously computing class logits over all pixels of all images in a batch, as is customary in Euclidean networks, is not directly applicable in hyperbolic space. This is because the explicit computation of the Möbius addition requires evaluating a tensor in $\mathbb{R}^{W \times H \times |\mathcal{Y}| \times d}$ for an images of size $(W \times H)$ with d embedding dimensions. Instead, they rewrite the Möbius addition as:

$$f_1 \oplus_c f_2 = \alpha f_1 + \beta f_2,$$
$$\alpha = \frac{1 + 2c\langle f_1, f_2 \rangle + c||f_2||^2}{1 + 2c\langle f_1, f_2 \rangle + c^2||f_1||^2||f_2||^2}, \qquad (18)$$
$$\beta = \frac{1 + c||f_1||^2}{1 + 2c\langle f_1, f_2 \rangle + c^2||f_1||^2||f_2||^2}.$$

This rewrite reduces the addition to adding two tensors in $\mathbb{R}^{W \times H \times |\mathcal{Y}|}$, allowing for per-pixel evaluation on image batches. For training, Ghadimi Atigh et al. (2022) incorporate hierarchical information by replacing the one-hot softmax with a hierarchical softmax:

$$p(\hat{y} = y|g(x)_{ij}) = \prod_{h \in \mathcal{H}_y} \frac{\exp(\xi_h(g(x)_{ij}))}{\sum_{s \in S_h} \exp(\xi_s(g(x)_{ij}))}, \qquad (19)$$

with $\mathcal{H}_y = \{y\} \cap \mathcal{A}_y$ the set containing y and its ancestors and S_h the set of siblings of class h. Performing per-pixel classification with hyperbolic hierarchical logistic regression opens up multiple new doors for image segmentation. First, the notion of uncertainty as given by the hyperbolic norm of output embeddings generalizes naturally to the pixel level. As shown in Fig. 4, the norm of pixel embeddings correlates with semantic ambiguity; the closer the pixel is to a semantic boundary the lower the pixel norm. Chen et al. (2022) have already used this insight to improve image segmentation. They outline a hyperbolic uncertainty loss, where the cross-entropy loss of a pixel is weighted as follows for pixel x_{ij}:
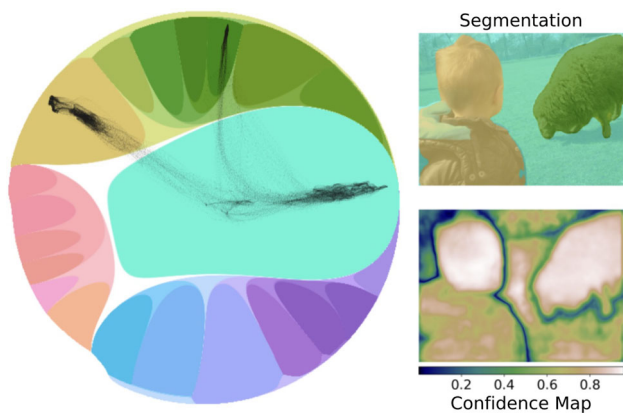
Fig. 4 Hyperbolic image segmentation naturally provides us per-pixel uncertainty information. Pixels with low hyperbolic norm constitute pixels with high uncertainty and are strongly correlated with closeness to semantic boundaries. Figure reproduced with permission of Ghadimi Atigh et al. (2022)

$$uw(x_{ij}) = 1 + \frac{1}{\log\left(t + \frac{d_h(g(x)_{ij},0)}{d_h(g(s),0)}\right)}, \qquad (20)$$

with s the most confident pixel and t a hyperparameter set to 1.02 in order to have a wide weight variation while avoiding division by zero. Adding this weight to the cross-entropy pixel loss consistently improves segmentation results for well-known segmentation networks. Other benefits of hyperbolic image segmentation include better zero-label generalization and higher effectiveness with few embedding dimensions compared to Euclidean pixel embeddings.

Hyperbolic kernel machines. Next to logistic regression, Cho et al. (2019) provide a general formulation for kernel methods in hyperbolic space with large-margin classifiers. Fang et al. (2021, 2023a) introduce positive definite kernel functions in hyperbolic space and show its potential for computer vision. Specifically, they propose hyperbolic instantiations of tangent kernels, radial basis function kernels, (generalized) Laplace kernels, and binomial kernels. The kernels can be plugged on top of convolutional networks and trained with cross-entropy to benefit from both the representation learning of the convolutional layers and the hyperbolic kernel dynamics in the classifier. Deep learning with hyperbolic kernel methods improves few-shot learning, person re-identification, and knowledge distillation. Zero-shot learning is even enabled through kernel distances between visual embeddings and semantic class representations.

3.2 Sample-to-Prototype Learning

The most popular strategy in hyperbolic learning is to represent classes as prototypes, i.e., as points in hyperbolic space. In this research direction, there are two solutions: embedding classes based on their sample mean, in the spirit of Prototypical Networks (ProtoNet) (Snell et al., 2017), or embeddings classes based on a given hierarchy over all classes.

Hyperbolic ProtoNet In Prototypical Networks (Snell et al., 2017), the prototype of a class k is determined as the mean vector of the samples belonging to that class:

$$P_{\mathbb{R}}(k) = \frac{1}{|S_k|} \sum_{y_s \in S_k} f_\theta(x_s), \qquad (21)$$

with S_k the set of samples belonging to class k. Inference can in turn be performed by assigning the label of the nearest prototype for a test sample. Khrulkov et al. (2020) generalize this formulation to Hyperbolic Prototypical Networks. Since computing averages in the Poincaré ball model requires expensive Fréchet mean calculations, they perform averaging using the Einstein midpoint, given in Klein coordinates as:

$$P_{\mathbb{K}}(k) = \sum_{i=1}^{|S_k|} \gamma_i g_{\mathbb{K}}(x_i) / \sum_{i=1}^{|S_k|} \gamma_i, \qquad (22)$$

with γ_i the Lorentz factors:

$$\gamma_i = \frac{1}{\sqrt{1 - c||g(x_i)||^2}}. \qquad (23)$$

Since Khrulkov et al. (2020) operate in the Poincaré ball model, this averaging operation requires transforming embeddings to and from the Klein model:

$$g_{\mathbb{K}}(x_i) = \frac{2g_{\mathbb{D}}(x_i)}{1 + c||g_{\mathbb{D}}(x_i)||^2},$$
$$g_{\mathbb{D}}(x_i) = \frac{g_{\mathbb{K}}(x_i)}{1 + \sqrt{1 - c||g_{\mathbb{K}}(x_i)||^2}}, \qquad (24)$$

with $g_{\mathbb{D}}(x_i)$ and $g_{\mathbb{K}}(x_i)$ the embeddings of input x_i in respectively the Poincaré ball model and the Klein model. Akin to its Euclidean counterpart, Hyperbolic ProtoNet is used to address few-shot learning, where the sample mean prototype serves as the class representation. Khrulkov et al. (2020) show that performing prototypical few-shot learning in hyperbolic space is competitive to Euclidean prototypical learning, even resulting in better accuracy scores when relying on a 4-layer ConvNet as the backbone.

As a follow-up work, Gao et al. (2021) show that different tasks and even individual classes in few-shot learning favor different curvatures. They propose to generate a per-class curvature based on the second-order statistics of its in-class and out-of-class sample representations. Using the second-order statistics, a multi-layer perceptron with sigmoid activation is

learned to fix the range of the curvature to [0, 1]. Given class-specific curvatures, prototypes are obtained by constructing an intra-class distance matrix on top of which an MLP is trained. The MLP serves as weights for each in-class sample. The procedure is repeated for the closest samples in the out-of-class set, after which the per-class prototype is given as the weighted hyperbolic average over the in-class and closest out-of-class samples. The curvature generation and weighted hyperbolic averaging improve few-shot learning in both inductive and transductive settings.

The hyperbolic clipping of Guo et al. (2022) is also effective for few-shot learning, consistently outperforming the standard ProtoNet and Hyperbolic ProtoNet on the CUB Birds and miniImageNet few-shot benchmarks. A few other works have extended Hyperbolic ProtoNet for few-shot learning with set- and grouplet-based learning and will be discussed in the sample-to-sample learning section.

Recently, Gao et al. (2022) investigate feature augmentation in hyperbolic space to solve the overfitting problem when dealing with limited data. On top, they introduce a scheme to estimate the feature distribution using neural-ODE. These elements are then plugged into few-shot approaches such as the hyperbolic prototypical networks of Khrulkov et al. (2020), improving performance. Choudhary & Reddy (2022) improve hyperbolic few-shot learning by reformulating hyperbolic neural networks through Taylor series expansions of hyperbolic trigonometric functions and show that it improves the scalability and compatibility, and outperforms Hyperbolic ProtoNet.

Hierarchical embedding of prototypes. Where Hyperbolic ProtoNets are effective in few-shot settings, a number of works have also investigated prototype-based solutions for the general classification. As starting point, these works commonly assume that the classes in a dataset are organized in a hierarchy, see Fig. 5. Long et al. (2020) embed action class hierarchy $\mathcal{H}$ in hyperbolic space using hyperbolic entailment cones (Ganea et al., 2018a), with an additional loss to increase the angular separation between leaf nodes to avoid inter-label confusion amongst class labels $\mathcal{Y}$. With $\mathcal{L}_H(\mathcal{H})$ as the hyperbolic embedding loss for hierarchy $\mathcal{H}$, let $\mathcal{P}$ denote the leave nodes of the hierarchy. Then the separation-based loss is given over the leaf nodes as:

$$\mathcal{L}_S(\mathcal{P}) = \mathbf{1}^T(\hat{P}\hat{P}^T - I)\mathbf{1}, \tag{25}$$

with $\hat{P}$ the ℓ_2-normalized representations of the leaf nodes. By combining the hierarchical and separation based losses, the hierarchy is embedded to balance both hierarchical constraints and discriminative abilities. The embedding is learned *a priori*, after which video embeddings are projected to the same hyperbolic space and optimized to their correct class embedding. This approach improves action recognition, zero-shot action classification, and hierarchical action
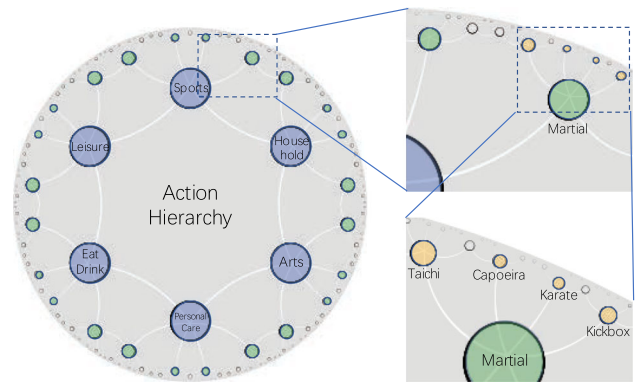
Fig. 5 Hierarchical knowledge amongst classes provides a structure for hyperbolic embeddings in computer vision approaches, where classes are represented as points or prototypes in hyperbolic space according to their hypernym-hyponym relations. For example, Long et al. (2020) exploit hierarchical relations from different actions for action hierarchies (right). Figure reproduced with permission of Long et al. (2020)

search. In a similar spirit, Dhall et al. (2020) show that using hyperbolic entailment cones for image classification is empirically better than using Euclidean entailment cones. Rather than separating hierarchical and visual embedding learning, Yu et al. (2022b) propose to simultaneously learn hierarchical and visual representations for skin lesion recognition in images. Image embeddings are optimized towards their correct class prototype, while the classes are optimized to abide by their hyperbolic entailment cones with an extra distortion loss to obtain better hierarchical embeddings. Gulshad et al. (2023) propose Hierarchical Prototype Explainer, a reasoning model in hyperbolic space to provide explainability in video action recognition. Their approach learns hierarchical prototypes at different levels of granularity e.g., parent and grandparent levels, to explain the recognized action in the video. By learning the hierarchical prototypes, they can provide explanations on different levels of granularity, including interpretation of the prediction of a specific class label and providing information on the spatiotemporal parts that contribute to the final prediction. Li et al. (2023c) investigate the semantic space of action recognition datasets and bridge the gap between different labeling systems. To achieve a unified action learning, actions are connected into a hierarchy using VerbNet (Schuler, 2005) and embedded as prototypes in hyperbolic space.

Hierarchical prototype embeddings have also been successfully employed in the zero-shot domain. Liu et al. (2020) show how to perform zero-shot learning with hyperbolic embeddings. Classes are embedded by taking their WordNet-based Poincaré Embeddings (Nickel & Kiela, 2017) and text-based Poincaré GloVe embeddings (Tifrea et al., 2019). Both are concatenated to obtain class prototypes. By optimizing seen training images to their prototypes, it becomes possible

to generalize to unseen classes during testing through a nearest neighbor search in the concatenated hyperbolic space. Xu et al. (2022) also perform hyperbolic zero-shot learning by training hyperbolic graph layers (Chami et al., 2019) on top of hyperbolic word embeddings. Dengxiong & Kong (2023) show the potential of hyperbolic space in generalized open set recognition, which classifies unknown samples based on side information. A side information (taxonomy) learning framework is introduced to embed the information in hyperbolic space with low distortion and identify the unknown samples. Moreover, an ancestor search algorithm is outlined to find the most similar ancestor in the taxonomy of the known classes.

For standard classification, Ghadimi Atigh et al. (2021) show how to integrate uniformity amongst prototypes in hyperbolic space by embedding classes with maximum separation on the boundary of the Poincaré ball given by Mettes et al. (2019); Kasarla et al. (2022). With prototypes now at the boundary of the ball, standard distance functions no longer apply since they are at the infinite distance to any point within the ball. To that end, they propose to use the Busemann distance, which is given for hyperbolic image embedding $g(x)$ and prototype p as:

$$b_p(g(x)) = \log\left(\frac{||p - g(x)||^2}{1 - ||g(x)||^2}\right). \tag{26}$$

By fixing prototypes with maximum separation *a priori* and minimizing this distance function with an extra regularization towards the origin, it becomes possible to perform hyperbolic prototypical learning with prototypes at the ideal boundary. Ghadimi Atigh et al. (2021) show that such an approach has direct links with conventional logistic regression in the binary case, highlighting its inherent properties. Moreover, maximally separated prototypes can also be replaced by prototypes from word embeddings or hierarchical knowledge, depending on the available knowledge and task at hand. In addition to standard classification, hierarchical hyperbolic embeddings have demonstrated effectiveness in continual learning (Gao et al., 2023). To learn the new data, Gao et al. (2023) propose a dynamically expanding geometry through a mixed-curvature space, enabling learning of complex hierarchies in a data stream. To prevent forgetting, angle-regularization and neighbor-robustness losses are used to preserve the geometry of the old data.

Few-shot learning has also been investigated with hierarchical knowledge. Zhang et al. (2022) perform such few-shot learning by first training a network on a joint classification and hierarchical consistency objective. The classification is given as a softmax over the class probabilities, as well as the softmax over the superclasses. In the few-shot inference stage, class prototypes are obtained through hyperbolic graph propagation to deal with the limited sample setting, improving few-shot learning as a result.

3.3 Sample-to-Sample Learning

Lastly, a number of recent works have investigated hyperbolic learning by contrasting between samples.

Hyperbolic Metric Learning Ermolov et al. (2022) investigate the potential of hyperbolic embedding for metric learning. In metric learning, the de facto solution is to match representations of sample pairs based on embeddings given by a pre-trained encoder. Rather than relying on Euclidean distances and contrastive learning for optimization, they propose a hyperbolic pairwise cross-entropy loss. Given a dataset with $|\mathcal{Y}|$ classes, each batch samples two samples from each category, i.e., $K = 2 \cdot |\mathcal{Y}|$. Then the loss function for a positive pair with the same class label is given as:

$$\ell_{ij} = -\log \frac{\exp(-D(g(x_i), g(x_j))/\tau)}{\sum_{k=1}^{K} \exp(-D(g(x_i), g(x_k))/\tau)}, \tag{27}$$

where $D(\cdot, \cdot)$ can be either a hyperbolic or a cosine distance and τ denotes a temperature hyperparameter. This loss is computed over all positive pairs (i, j) and (j, i) in a batch. Using supervised (Dosovitskiy et al., 2021) and self-supervised (Caron et al., 2021) vision transformers as encoders, hyperbolic metric learning consistently outperforms Euclidean alternatives and sets state-of-the-art on fine-grained datasets.

Hyperbolic metric learning has shown to be effective to overcome overfitting and catastrophic forgetting in few-shot class-incremental learning tasks, explored by Cui et al. (2022). This is done by adding a metric learning loss as a part of the distillation in continual learning. They also propose a hyperbolic version of Reciprocal Point Learning (Chen et al., 2020a) to provide extra-class space for known categories in the few-shot learning stage. Yan et al. (2023) also explore hyperbolic metric learning, incorporating noise-insensitive and adaptive hierarchical similarity to handle noisy labels and multi-level relations. Kim et al. (2022) add a hierarchical regularization term on top of the metric learning approaches, with the goal of learning hierarchical ancestors in hyperbolic space without any annotation. Hyperbolic metric learning is furthermore effective in semantic hashing (Amin et al., 2022), face recognition via large-margin nearest-neighbor learning (Trpin & Boshkoska, 2022), and multi-modal alignment given videos and knowledge graph (Guo et al., 2021).

Following the progress of large language models and the success of vision-language models (e.g., CLIP (Radford et al., 2021)) in multimodal representation learning, Desai et al. (2023) propose a hyperbolic image-text representation in Lorentz model. The proposed method first processes the input image and text using two separate encoders. Then, the generated embedding is projected into the hyperbolic space, and training is performed using a contrastive and entailment loss.

The paper shows that the proposed approach outperforms the Euclidean CLIP as it is capable of capturing hierarchical multimodal relations in hyperbolic space. Hong et al. (2023b) also explore multimodal data to perform zero-shot learning with audio-visual data with a curvature-aware geometric solution. To align the features extracted from the audio and video modalities, Hong et al. (2023b) propose *Hyper-align*, a hyperbolic alignment loss in a fixed curvature setup, followed by *Hyper-single*, a module to enable learnable curvature, and *Hyper-multiple*, to calculate the alignment loss within different curvatures.

Hyperbolic set-based learning. Where sample-to-prototype and sample-to-sample approaches compare samples to individual elements, some works have shown that set-based and group-based distances are more effective and robust. Ma et al. (2022) introduce an adaptive sample-to-set distance function in the context of few-shot learning. Rather than aggregating support samples to a single prototype, an adaptive sample-to-set approach is proposed to increase the robustness to the outliers. The sample-to-set function is a weighted average of the distance from the query to all support samples, where the distance is calculated with a small network over the feature maps of the query and support samples. This approach

benefits few-shot learning, especially when dealing with outliers.

In the context of metric learning, Zhang et al. (2021a) argue that sample-to-sample learning is computationally expensive, while sample-to-prototype learning is less accurate. They propose a hybrid strategy based on grouplets. Each grouplet is a random subset of samples and the set of grouplets is matched with prototypes through a differentiable optimal transport. Akin to Ermolov et al. (2022), they show that using hyperbolic embedding spaces improved metric learning on fine-grained datasets. Moreover, they provide empirical evidence that other metric-based losses benefit from hyperbolic embeddings, highlighting the general utility of hyperbolic space for metric learning.

4 Unsupervised Hyperbolic Visual Learning

Hyperbolic learning has been actively researched in the unsupervised domain of computer vision. We identify three dominant research directions in which hyperbolic deep learning has found success: generative learning, clustering, and self-supervised learning. Below, each is discussed separately.
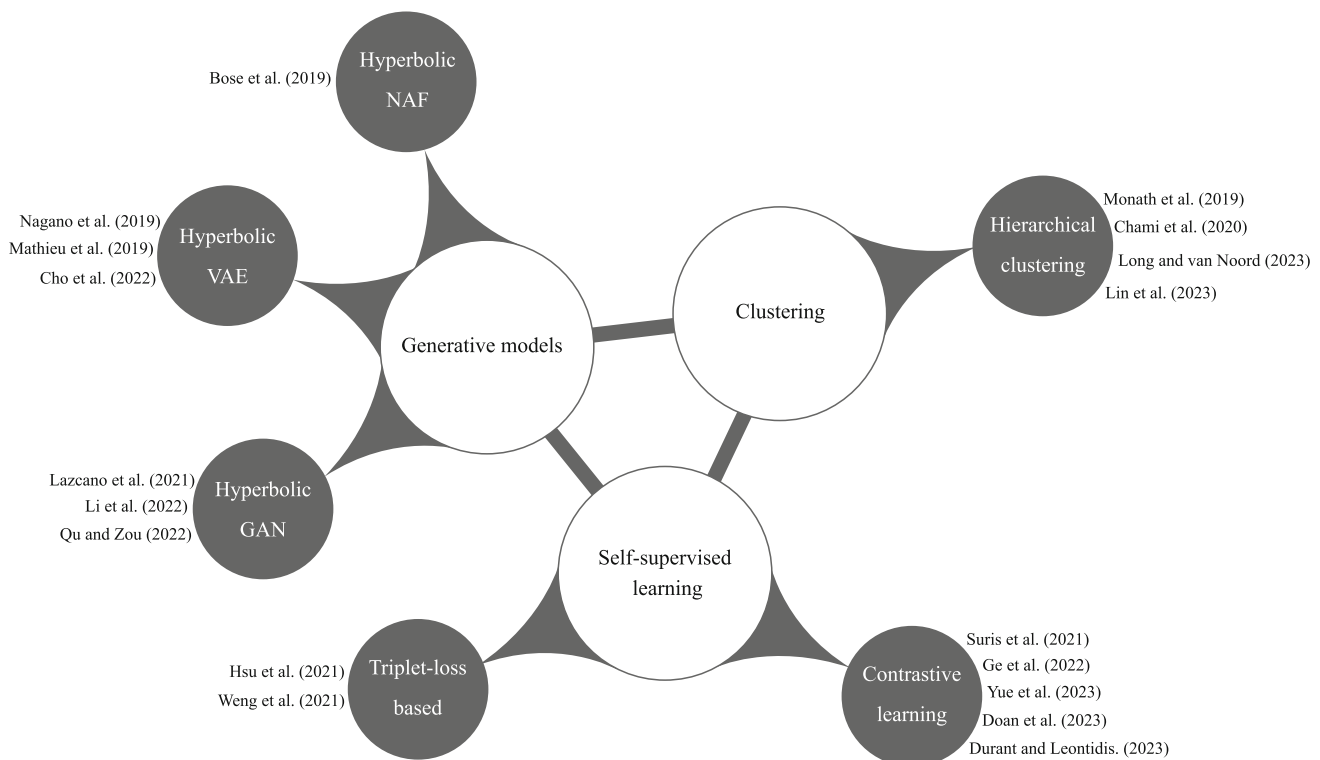
Fig. 6 The three major methods for unsupervised hyperbolic learning in computer vision. Current literature performs unsupervised learning in hyperbolic space using (i) generative models, (ii) clustering, (iii) self-supervised learning

4.1 Generative Approaches

4.1.1 Hyperbolic VAEs

Variational autoencoders (VAEs) (Kingma & Welling, 2013; Rezende et al., 2014) with hyperbolic latent space have been used to learn representations of images. Nagano et al. (2019) propose the hyperbolic wrapped normal distribution in Lorentz model and derive algorithms for both reparametrizable sampling and computing the probability density function. They then derive a hyperbolic β-VAE (Higgins et al., 2017) using the wrapped normal function as the prior and posterior, replacing the usual (Euclidean) Gaussian distribution. The wrapped normal distribution in a manifold $\mathcal{M}$ is the pushforward measure under the exponential map $\exp_{\mathcal{M}}$. Thus, a sample z can be obtained as (Mathieu et al., 2019):

$$z = \exp_{\mu}^{\mathcal{M}}\left(G(\mu)^{-1/2}v\right), \, v \sim \mathcal{N}(\cdot|0, \Sigma) \quad (28)$$

where $\exp_{\mu}^{\mathcal{M}}$ is the exponential map of $\mathcal{M}$ at μ and G is the matrix representation of the metric of $\mathcal{M}$, and v is a random sample from Euclidean normal distribution with mean 0 and variance Σ. To accommodate the geometry of the latent space, exponential and logarithmic maps were added at the end of the VAE encoder and before the start of the VAE decoder, respectively. In order to train their hyperbolic VAE with the typical evidence lower bound, Nagano et al. (2019) compute the density of the wrapped normal distribution using the change-of-variables formula. Since their sampling algorithm required the exponential and parallel transport maps, Nagano et al. (2019) compute the log-determinants and inverses of these maps in order to apply the change-of-variables formula. Nagano et al. (2019) then use their VAE to learn representations of MNIST and Atari 2600 Breakout screens. On MNIST, Hyperbolic representations outperform Euclidean representations at low latent dimensions but were overtaken starting at dimension 10.

Mathieu et al. (2019) extend the work of Nagano et al. (2019) by introducing the Riemannian normal distribution and deriving reparametrizable sampling schemes for both the Riemannian normal and wrapped normal using hyperbolic polar coordinates. The Riemannian normal views the Euclidean normal distribution as the distribution minimizing the entropy for a given mean and standard deviation and defines a new normal distribution on hyperbolic space with this property:

$$\mathcal{N}_{\mathcal{M}}^R(z|\mu, \sigma^2) = \frac{1}{Z^R}\exp\left(-\frac{d_{\mathcal{M}}(\mu, z)^2}{2\sigma^2}\right) \quad (29)$$
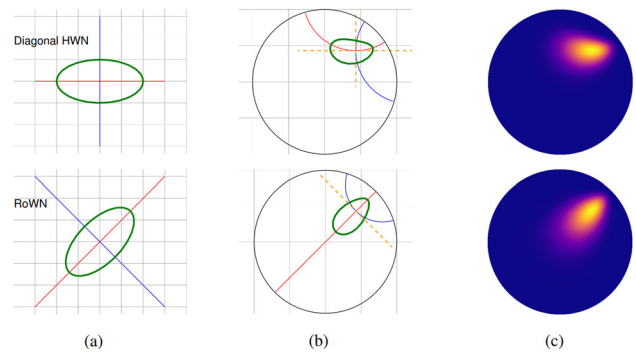
Fig. 7 The **standard hyperbolic wrapped normal** (top) and **rotated hyperbolic wrapped normal** (bottom). In (**a**), the principal axes of the normal distribution are illustrated. In (**b**), the principal axes of the transported normal distribution are visualized. The density of the two distributions are visualized in (**c**). Image courtesy of Cho et al. (2022)

where Z^R is a normalizing constant, μ and σ^2 are the mean and variance. Mathieu et al. (2019) additionally introduce the use of a gyroplane layer as the first layer of the decoder, following Ganea et al. (2018b). Noting that a Euclidean affine transform can be written as

$$f_{a,p}(z) = \text{sign}(\langle a, z - p \rangle)||a||d_E(z, H_{a,p})$$

where $H_{a,p} = \{z \in \mathbb{R}^n | \langle a, z - p \rangle = 0\}$ is the decision hyperplane, they replace each piece of the formula with its hyperbolic counterpart to obtain

$$f_{a,p}^c(z) = \text{sign}(\langle a, \log_p^c(z) \rangle_p)||a||_p d_p^c(z, H_{a,p}^c) \quad (30)$$

where all $H_{a,p}^c = \{z \in \mathbb{H} | \langle a, \log_p^c(z) \rangle = 0\}$. The closed-form formula for the distance term in the Poincaré ball is

$$d_p^c(z, H_{a,p}^c) = \frac{1}{\sqrt{c}}\sinh^{-1}\left(\frac{2\sqrt{c}|\langle -p \oplus_c z, a \rangle|}{(1 - c||-p \oplus_c z||^2)||a||}\right) \quad (31)$$

Mathieu et al. (2019) also use their hyperbolic VAE to learn representations of MNIST and find that using both the Riemannian normal and the gyroplane layer improve test log-likelihoods, especially at low latent dimensions.

Cho et al. (2022) extend the previous two works by proposing a new version of the hyperbolic wrapped normal distribution (HWN) in Lorentz model. Their primary observation is that for the wrapped normal distribution, the principal axes of the distributions are not aligned with the local standard axes, see Fig. 7. They propose a new sampling process that fixes the alignment of the principal axes, resulting in a new distribution which they call the rotated hyperbolic wrapped normal (RoWN). Given a mean μ in the Lorentz model of hyperbolic geometry and a diagonal covariance matrix Σ, samples from the RoWN distribution are sampled as follows:

1. Find the rotation matrix R that rotates the x-axis $x = ([\pm 1, \ldots, 0])$ to $y = \mu_{1:}$. We can compute R as

$$R = I + (y^{\mathsf{T}}x - x^{\mathsf{T}}y) + \frac{(y^{\mathsf{T}}x - x^{\mathsf{T}}y)^2}{1 + \langle x, y \rangle} \qquad (32)$$

2. Rotate Σ by R: $\hat{\Sigma} = R\Sigma R^{\mathsf{T}}$
3. Now sample as in the usual hyperbolic wrapped normal: sample $v \sim \mathcal{N}(\mathbf{0}, \hat{\Sigma})$ and then map it to hyperbolic space as follows: $\exp_{\mu}(\mathrm{PT}_{\mathbf{0}\to\mu}([0, v]))$

Cho et al. (2022) find that RoWN outperforms HWN in a variety of settings, such as the Atari 2600 Breakout image generation experiment first examined in Nagano et al. (2019).

4.1.2 Hyperbolic GANs

Using the intuition that images are organized hierarchically, several works have proposed hyperbolic generative adversarial networks (GANs). Lazcano et al. (2021) propose a hyperbolic GAN which replaces some of the Euclidean layers in both the generator and discriminator with hyperbolic layers (Ganea et al., 2018a) with learnable curvature. Lazcano et al. (2021) propose hyperbolic variants of the original GAN (Goodfellow et al., 2020), the Wasserstein GAN WGAN-GP (Gulrajani et al., 2017) and conditional GAN CGAN (Mirza & Osindero, 2014). The paper finds that their best configurations of Euclidean and hyperbolic layers generally improved the Inception Score (Salimans et al., 2016) and Frechet Inception Distance (Heusel et al., 2017) on MNIST image generation, with the best improvements in the GAN architecture. The best learned curvatures are close to zero. Unlike other hyperbolic generative models (VAEs and normalizing flows), good results are observed at large latent dimensions.

Qu & Zou (2022a) propose HAEGAN, a hyperbolic autoencoder and GAN framework in the Lorentz model $\mathbb{L}$ (also known as the hyperboloid model), of hyperbolic geometry. The GAN is based on the structure of WGAN-GP (Arjovsky et al., 2017; Gulrajani et al., 2017). The structure of HAEGAN consists of an encoder, which takes in real data and generates real representations, and a generator, which takes in noise and generates fake representations. A critic is trained to distinguish between the two representations, and a decoder takes the fake representations and produces the final generated object. Qu & Zou (2022a) generalize WGAN-GP to hyperbolic space using three operations: the first is the hyperbolic linear layer is $\mathrm{HLinear}_{n,m} : \mathbb{L}^n_K \to \mathbb{L}^m_K$ of Chen et al. (2021), the second the hyperbolic centroid distance layer $\mathrm{HCDist}_{n,m}(x) : \mathbb{L}^n_K \to \mathbb{R}^m$ of Liu et al. (2019),
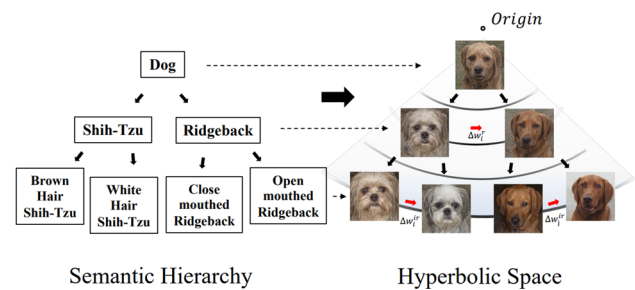
Fig. 8 Hierarchical attribute editing in hyperbolic space is possible due to hyperbolic space's ability to encode semantic hierarchical structure within image data. Changing the high-level, category-relevant details (closest to the origin) changes the category, while changing low-level (farthest from the origin), category-irrelevant attributes varies images within categories. Image courtesy of Li et al. (2023b)

and the third a a new Lorentz concatenation layer:

$$\mathrm{HCat}\left(\{x_i\}_{i=1}^N\right) = \left[\sqrt{\sum_{i=1}^N x_{i_t}^2 + (N-1)/K}, \, x_{1_s}^{\mathsf{T}}, \ldots, x_{1_s}^{\mathsf{T}}\right]^{\mathsf{T}} \qquad (33)$$

Compared to previous work (Shimizu et al., 2021), the HCat layer has the advantage of always having bounded gradients (Shimizu et al., 2021). Compared to Lazcano et al. (2021), HAEGAN shows improved results on MNIST image generation.

Li et al. (2023b) propose a hyperbolic method for few-shot image generation. The main idea is that hyperbolic space encodes a semantic hierarchy, where the root of the hierarchy (i.e., at the center of hyperbolic space) is a category, e.g., dog. At lower levels, we have more fine-grained separations, such as subcategories, e.g., Shih-Tzu and Ridgeback dogs. Finally, at the lowest level, there are category-irrelevant features, e.g., the hair color or pose of the dog (see Fig. 8). This method builds on the Euclidean pSp method (Richardson et al., 2021) for image-to-image translation. The pSp method uses a feature pyramid to extract feature maps and uses a set of projection heads on these feature maps to produce each of the style vectors required by StyleGAN (Karras et al., 2019, 2020), which is commonly denoted the $\mathcal{W}^+$-space. Image-to-image translation can then be done by editing or replacing style vectors. Li et al. (2023b) generalize to hyperbolic space by mapping the output of a frozen, pre-trained pSp encoder to hyperbolic space and then back to the $\mathcal{W}^+$-space of style vectors, and then feeding the style vectors into a frozen, pre-trained StyleGAN. Projection to hyperbolic space is done using the Mobius layer $f^{\otimes c}$ of Ganea et al. (2018b), with the full projection layer having the form

$$z_{\mathbb{D}i} = f^{\otimes c}(\exp_0^c(\mathrm{MLP}_E(\mathbf{w}_i))) \qquad (34)$$

with mapping back to the $\mathcal{W}^+$-space achieved by a logarithmic map plus an MLP. Li et al. (2023b) supervise the hyperbolic latent space with a hyperbolic classification loss based on the multinomial logistic regression formulation of Ganea et al. (2018b). After calculating the probabilities, the loss function is just negative log-likelihood as

$$\mathcal{L}_{\text{hyper}} = -\frac{1}{N} \sum_{i=1}^{N} \log(p_n) \tag{35}$$

The full loss function is the pSp loss function plus this term, excluding a specific facial reconstruction loss used by the pSp method, since Li et al. (2023b) do not focus on face generation. Li et al. (2023b) perform image generation as follows: given an image x_i, the image is embedded in hyperbolic space with representation $g_{\mathbb{D}}(x_i)$ and is rescaled to the desired radius (i.e., fine-grained-ness) r. A random vector is then sampled from the seen categories and a point is taken on the geodesic between the two points. Li et al. (2023b) find that their method is competitive with state-of-the-art methods and show promise for image-to-image transfer.

4.1.3 Hyperbolic Normalizing Flows

Bose et al. (2020) propose a hyperbolic normalizing flow in Lorentz model that generalizes the Euclidean normalizing flow RealNVP (Dinh et al., 2016) to hyperbolic space. They propose two types of hyperbolic normalizing flows: the first, which they call tangent coupling, which carries out the coupling layer of RealNVP in the tangent space at the hyperbolic origin o:

$$\tilde{f}^{\mathcal{TC}}(\tilde{x}) = \begin{cases} \tilde{z}_1 = \tilde{x}_1 \\ \tilde{z}_2 = \tilde{x}_2 \odot \sigma(s(\tilde{x}_1)) + t(\tilde{x}_1) \end{cases} \tag{36}$$

$$f^{\mathcal{TC}}(x) = \exp_o^K(\tilde{f}^{\mathcal{TC}}(\log_o^K(x))) \tag{37}$$

where s, t are neural networks and σ is a pointwise non-linearity.

The wrapped hyperboloid extends tangent coupling by using parallel transport to map intermediate vectors from the tangent space of the origin to the tangent space of another point in hyperbolic space:

$$\tilde{f}^{\mathcal{WHC}}(\tilde{x}) = \begin{cases} \tilde{z}_1 = \tilde{x}_1 \\ \tilde{z}_2 = \log_o^K\left(\exp_{t(\tilde{x}_1)}^K\left(\text{PT}_{o \to t(\tilde{x}_1)}(v)\right)\right) \end{cases} \tag{38}$$

$$v = \tilde{x}_2 \odot \sigma(s(\tilde{x}_1)) \tag{39}$$

$$f^{\mathcal{WHC}}(x) = \exp_o^K(\tilde{f}^{\mathcal{WHC}}(\log_o^K(x))) \tag{40}$$

Compared to tangent coupling, wrapped hyperbolic coupling allows the flow to leverage different parts of the manifold

instead of just the origin. The paper also derives the inverse and Jacobian determinants of the two flows. As is the case for hyperbolic VAEs, Bose et al. (2020) also benchmark on MNIST, and find a similar trend as Nagano et al. (2019): the performance of hyperbolic models exceed that of the equivalent Euclidean model at low dimension, but as early as latent dimension 6 Euclidean models overtake hyperbolic models in performance. Bose et al. (2020) find that hyperbolic normalizing flows outperform hyperbolic VAEs at these low latent dimensions.

4.2 Clustering

Due to the close relationship between hyperbolic space, hierarchies, and trees, several works have explored hierarchical clustering using hyperbolic space. Monath et al. (2019) propose to perform hierarchical clustering using hyperbolic representations. Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^N$, Monath et al. (2019) require a hyperbolic representation at the edge of the Poincaré disk $\mathbb{D}^d$ for each data point $x_i \in \mathcal{D}$, which becomes the leaves of the hierarchical clustering. The method of Monath et al. (2019) creates a hierarchical clustering by optimizing the hyperbolic representations for a fixed number of internal nodes. Parent–children dissimilarity between a child representation z_c and a parent representation z_p is measured by

$$d_{cp}(z_c, z_p) = d_{\mathbb{D}}(z_c, z_p)(1 + \max\{||z_p||_{\mathbb{D}} - ||z_c||_{\mathbb{D}}, 0\}) \tag{41}$$

which encourages children to have larger norms than their parents. A discrete tree can then be extracted as follows:

$$\text{Parent}(z_c) = \arg\min_{||z_p|| < ||z_c||} d_{cp}(z_c, z_p) \tag{42}$$

The internal node observations are supervised by two losses: first, a hierarchical clustering loss based on Dasgupta's cost (Dasgupta, 2016) and a continuous extension due to Wang & Wang (2018) that reformulates the loss in terms of lowest common ancestors (LCAs), and second, a parent–child margin objective that encourages parent nodes to have smaller norm than their children.

Suppose $\mathcal{D}$ has pairwise similarities $\{w_{ij}\}_{i,j \in [N]}$. A hierarchical clustering of $\mathcal{D}$ is a rooted tree T such that each leaf is a data point. For leaves $i, j \in T$, denote their LCA by $i \vee j$, the subtree rooted at $i \vee j$ by $T[i \vee j]$, and the leaves of $T[i \vee j]$ by $\text{leaves}(T[i \vee j])$. Finally, let relation $\{i, j|k\}$ holds if $i \vee j$ is a descendant of $i \vee j \vee k$. Then Dasgupta's cost can be formulated as

$$C_{\text{Dasgupta}}(T; w) = \sum_{ij} w_{ij} |\text{leaves}(T[i \vee j])| \tag{43}$$

Wang & Wang (2018) show that

$$C_{\text{Dasgupta}}(T; w) = \sum_{ijk} [w_{ij} + w_{ik} + w_{jk} - w_{ijk}(T; w)]$$
$$+ 2 \sum_{ij} w_{ij} \tag{44}$$

where

$$w_{ijk}(T; w) = w_{ij} \mathbb{1}[\{i, j | k\}] + w_{ik} \mathbb{1}[\{i, k | j\}]$$
$$+ w_{jk} \mathbb{1}[\{j, k | i\}] \tag{45}$$

The margin parent–child dissimilarity is given as

$$d_{cp}(z_c, z_p; \gamma) = d_{\mathbb{D}}(z_c, z_p)(1 + \max\{||z_p||_{\mathbb{D}}$$
$$- ||z_c||_{\mathbb{D}} + \gamma, 0\}) \tag{46}$$

and the total margin objective is

$$\mathcal{L}_{cp} = \sum_{z_c} d_{cp}(z_c, \texttt{Parent}(z_c); \gamma) \tag{47}$$

The embedding is alternately optimized between the clustering objective and the parent–child objective. Optimization of the hyperbolic parameters is done via the method of Nickel & Kiela (2017). Using this method, Monath et al. (2019) are able to embed ImageNet using representations taken from the last layer of a pre-trained Inception neural network.

Similar to Monath et al. (2019), Chami et al. (2020a) base their method on Dasgupta's cost (Eq. 43) and Wang and Wang's (Eq. 44) reformulation in terms of LCAs. Chami et al. (2020a) define the LCA of two points in hyperbolic space to be the point on the geodesic connecting the two points that are closest to the hyperbolic origin, and provide a formula to calculate this point in the Poincaré disk $\mathbb{D}$. This formula allows Eq. 44 to be directly optimized by replacing the $w_{ijk}(T; w)$ terms with its continuous counterpart. A hierarchical clustering tree can then be produced by iteratively merging the most similar pairs, where similarity is measured by their hyperbolic LCA distance from the origin. Unlike the method of Monath et al. (2019), Chami et al. (2020a) do not require hyperbolic embeddings to be available, and optimize the hyperbolic embeddings of the whole tree, not just the leaves.

Recently, Long & van Noord (2023) propose a scalable Hyperbolic Hierarchical Clustering (sHHC) enabling learning of a continuous hierarchy which is also scalable to large datasets. They use clustering to extract hierarchical pseudo-labels from sound and vision and perform a downstream cross-modal self-supervised task, achieving competitive performance. They augment the hyperbolic clustering of Chami et al. (2020a) by pre-clustering of the data point features into

evenly-sized clusters is performed using the Sinkhorn fixed-point iteration method of Asano et al. (2019). Hyperbolic clustering is then performed using the method of Chami et al. (2020a). Finally, the clustering is self-supervised using the method of Long et al. (2020).

Lin et al. (2023a) propose a neural-network based framework for the hierarchical clustering of multi-view data. The framework consists of two steps: first, improving representation quality via reconstruction loss, contrastive learning between different views, and a weighted triplet loss between positive examples and mined hard negative examples, and second, applying the hyperbolic hierarchical clustering framework of Chami et al. (2020a).

The contrastive loss in Lin et al. (2023a) is the usual contrastive loss (see following section) where positive examples are views from the same object and negative examples are views from different objects. The weighted triplet loss is

$$\mathcal{L}_m = \frac{1}{N} \sum_{i=1}^N w^m(a_i, p_i)[m + ||a_i - p_i||_2^2 - ||a_i - n_i||_2^2]_+ \tag{48}$$

where a_i refer to the anchor points, p_i are the positive examples, and n_i are the negative examples. Positive and negatives examples are mined based on the method of Iscen et al. (2017), which measures the similarity of a pair of points based on estimating the data manifold using k-nearest neighbors graphs. Lin et al. (2023b) apply their method to perform multi-view clustering for a variety of multi-view image datasets.

4.3 Self-Supervised Learning

In Sect. 4.3.1, we describe methods for hyperbolic self-supervision that are primarily based on triplet losses, and in Sect. 4.3.2 we discuss methods for hyperbolic self-supervision which are primarily based on contrastive losses.

4.3.1 Hyperbolic Self-Supervision

Based on the idea that biomedical images are inherently hierarchical, Hsu et al. (2021) propose to learn patch-level representations of 3D biomedical images using a 3D hyperbolic VAE and to perform 3D unsupervised segmentation by clustering the representations. Hsu et al. (2021) extend the hyperbolic VAE architecture of Mathieu et al. (2019) using a 3D convolutional encoder and decoder as well as gyroplane convolutional layer that generalizes the Euclidean convolution with the gyroplane layer of Ganea et al. (2018b) (See Eqs. 30 and 31). In order to learn good representations, the paper proposes to use a hierarchical self-supervised loss that

captures the implicit hierarchical structure of 3D biomedical images.

To capture the hierarchical structure of 3D biomedical images, Hsu et al. (2021) propose that given a parent patch μ_p, to sample a child patch μ_c which is a subpatch of the parent patch, and a negative patch μ_n that does not overlap with the parent patch. Then the hierarchical self-supervised loss is defined as a margin triplet loss as follows:

$$\mathcal{L}_{\text{hierarchical}} = \max(0, d_{\mathbb{D}}(\mu_p, \mu_c) - d_{\mathbb{D}}(\mu_p, \mu_n) + \gamma) \quad (49)$$

This encourages the representations of subpatches to be children or descendants of the representation of the main patch, and faraway patches (which likely contain different structures) to be on other branches of the learned hierarchical representation.

To perform unsupervised segmentation, the learned latent representations are extracted and clustered using a hyperbolic k-means algorithm, where the traditional Euclidean mean is replaced with the Frechet mean. For a manifold $\mathcal{M}$ with metric $d_{\mathcal{M}}$, the Frechet mean of a set of points $\{z_i\}_{i=1}^k, z_i \in \mathcal{M}$ is defined as the point μ that minimizes the squared distance to all points z_i:

$$\mu_{\text{Fr}} = \arg\min_{\mu \in \mathcal{M}} \frac{1}{k} \sum_{i=1}^k d_{\mathcal{M}}(z_i, \mu)^2 \quad (50)$$

and is one way to generalize the concept of a mean to manifolds. Unfortunately, the Frechet mean on the Poincaré ball does not admit a closed-form solution, so Hsu et al. (2021) compute the Frechet mean with the iterative algorithm of Lou et al. (2020). The paper finds that this strategy is effective for the unsupervised segmentation of both synthetic biological data and 3D brain tumor MRI scans (Menze et al., 2014; Bakas et al., 2017, 2018).

Weng et al. (2021) propose to leverage the hierarchical structure of objects within images to perform weakly-supervised long-tail instance segmentation. To capture this hierarchical structure, Weng et al. (2021) learn hyperbolic representations which are supervised with several hyperbolic self-supervised losses. Instance segmentation is done in three stages: first, mask proposals are generated using a pre-trained mask proposal network. Mask proposals consists of bounding boxes $\{\mathcal{B}_i\}_{i=1}^k$ and masks $\{\mathcal{M}_i\}_{i=1}^k$. Define x_i^{full} to be the original image cropped to bounding box $\mathcal{B}_i$, x_i^{bg} to be the cropped image with the object masked out using mask $1 - \mathcal{M}_i$, and x_i^{fg} to be the same cropped image with the background masked out using mask $\mathcal{M}_i$. We will refer to these as the full object image, object background, and object, respectively.

Second, hyperbolic representations of $z_i^{\text{bg}} = g(x_i^{\text{bg}})$, and $z_i^{\text{fg}} = g(x_i^{\text{fg}})$ are learned by a pre-trained feature extractor and

supervised by a combination of three self-supervised losses. The representations are fixed to have latent dimension 2. The first self-supervised loss encourages the representation of the object to be similar to that of the full object image and farther away from the representation of the object background:

$$\mathcal{L}_{\text{mask}} = \sum_{i=1}^k \max(0, \gamma - d(z_i^{\text{full}}, z^{\text{fg}}) + d(z_i^{\text{full}}, z_i^{\text{bg}})) \quad (51)$$

The second loss is a triplet loss that requires the sampling of positive and negative examples.

$$\mathcal{L}_{\text{object}} = \sum_{i=1}^k \max(0, \gamma - d(z_i^{\text{fg}}, \hat{z}^{\text{fg}}) + d(z_i^{\text{fg}}, \overline{z}_i^{\text{fg}})) \quad (52)$$

where $\hat{z}^{\text{fg}}$ and $\overline{z}_i^{\text{fg}}$ are the features of the positive and negative samples.

The third loss is similar to the hierarchical triplet loss of Hsu et al. (2021) described above, except with the origin taking the place of negative samples:

$$\mathcal{L}_{\text{hierarchical}} = \sum_{i=1}^k \max(0, \gamma - d(z_i^{\text{child}}, o) - d(z_i^{\text{fg}}, o)) \quad (53)$$

where o represents the origin of the Poincaré ball, and z_i^{child} is the feature of the child mask of proposal i.

Finally, the representations are clustered using hyperbolic k-means clustering. Unlike (Hsu et al., 2021), to compute the mean they map the representations from the Poincaré disk to the hyperboloid model $\mathcal{L}$ and compute the (weighted) hyperboloid midpoint proposed by Law et al. (2019):

$$\mu = \sqrt{\beta} \frac{\sum_{i=1}^k v_i x_i}{\left| \|\sum_{i=1}^k v_i x_i\|_{\mathcal{L}} \right|} \quad (54)$$

where β is $-1/\text{curvature}$.

Compared to the Frechet mean, this mean has the advantage of having a closed-form formula, making it more computationally efficient. Weng et al. (2021) find that their method improves other partially-supervised methods on the LVIS long-tail segmentation dataset (Gupta et al., 2019).

4.3.2 Hyperbolic Contrastive Learning

Hyperbolic contrastive learning methods have also been proposed. Surís et al. (2021) propose to learn hyperbolic representations for video action prediction because of their ability to combine representing hierarchy and giving a measure of uncertainty (see Fig. 9). Surís et al. (2021) learn an action hierarchy where more abstract actions are near the origin of the Poincaré disk and more fine-grained actions are
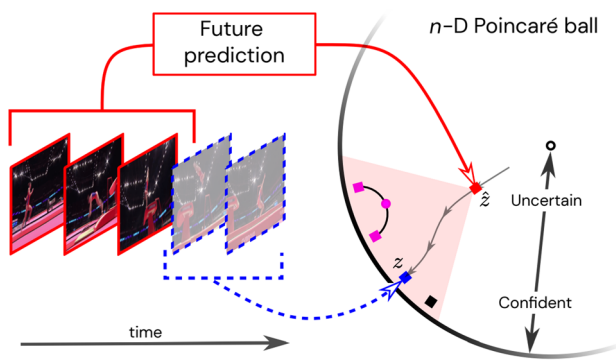
Fig. 9 Surís et al. (2021) model uncertainty with hyperbolic representations. If the model is uncertain, it can predict an abstraction of all possible actions (red square), and if it is certain it can predict a more specific action (blue square). The pink circle shows how computing the mean of two representations (pink squares) increases the generality. Figure reproduced with permission of Surís et al. (2021)

near the edge. If the preceding video frames are ambiguous, this hierarchical representation allows the ability to predict a more general parent category of action (e.g., greeting) instead of having to predict more fine-grained child categories of action (e.g., handshake or high-five). The parent of two actions is computed as the hyperbolic mean of their hyperbolic representations, which Surís et al. (2021) compute as the midpoint of the geodesic connecting the two representations. Surís et al. (2021) propose a two-stage framework for video action prediction which consists first of contrastive pre-training hyperbolic representations, then freezing the representations and training a linear classifier for action prediction.

Self-supervised pre-training proceeds as follows: let x_t be a frame of the video, and a representation $z_t = f(x_t)$ is produced by an encoder f. The pretext task is to predict the representation $z_{t+\delta}$ of a clip δ frames into the future. The model produces an estimate $\hat{z}_{t+\delta} = \phi(c_t, \delta)$, where $c_t = g(z_1, \ldots, z_t)$ is an encoding of all past video frames. All function f, g, ϕ are parameterized by a neural network. The training is supervised by a contrastive loss:

$$\mathcal{L} = -\sum_i \left[\log \frac{\exp(-d_{\mathbb{D}}^2(\hat{z}_i, z_i))}{\sum_j \exp(-d_{\mathbb{D}}^2(\hat{z}_i, z_j))} \right] \quad (55)$$

which encourages the positive pairs $\hat{z}_i, z_i$ to have similar representations while pushing $\hat{z}_i$ from the representations of all negative examples z_j. One key feature of this loss is that under the presence of uncertainty, say when actions a, b are probable, $\mathcal{L}$ is minimized by predicting the midpoint on the geodesic connecting a, b, which is equivalent to moving one level up the hierarchy to the parent of a, b.

Ge et al. (2023) propose to improve contrastive learning by incorporating the hierarchical structure of images with a scene-object hierarchy (see Fig. 10). Ge et al. (2023) use a

Fig. 10 The learned hierarchy of Ge et al. (2023) has objects near the origin of the Poincaré disk and scenes near the edge of hyperbolic space. Image courtesy of Ge et al. (2023)

hyperbolic version of the MoCo architecture He et al. (2020), which the authors call HCL. Ge et al. (2023) extend the MoCo architectures in several ways: first, unlike previous works for visual contrastive learning, HCL requires that object regions be extracted from the input image. Secondly, a hyperbolic backbone along with a corresponding momentum encoder is added to MoCo's Euclidean backbone and its momentum encoder. The Euclidean backbone and momentum encoder are trained the same way as in He et al. (2020), but the inputs are not images but the extracted object regions. The hyperbolic branch takes as input a scene region u and an object region v that is a subregion of the scene u, and negative objects $\mathcal{N}_u = \{n_1, \ldots, n_k\}$ that are not subregions of the scene u. Let the representations of u, v, n_j be z_u, z_v, z_j, respectively. The hyperbolic branch is then trained with a contrastive loss with hyperbolic distance as the similarity measure:

$$\mathcal{L}_{\text{hyp}} = -\log \frac{\exp\left(-\frac{d_{\mathbb{D}}(z_u, z_v)}{\tau}\right)}{\exp\left(-\frac{d_{\mathbb{D}}(z_u, z_v)}{\tau}\right) + \sum_j \exp\left(-\frac{d_{\mathbb{D}}(z_u, z_j)}{\tau}\right)} \quad (56)$$

where τ is a temperature parameter. This loss encourages representations to form a scene-object hierarchy where scenes have the highest norm (i.e., are at the edge of the Poincaré ball $\mathbb{D}$) and objects have the smallest norm (i.e., are at the center of $\mathbb{D}$). The paper finds that their method achieves small gains over the original MoCo and MoCo augmented with bounding box information. They also examine the repre-

sentations of out-of-context objects using their method, and find that they generally have higher distance to the scene images.

Yue et al. (2023) propose a different method for hyperbolic contrastive learning that is based on SimCLR (Chen et al., 2020c). Like Ge et al. (2023), Yue et al. (2023) replace the dot-product similarity of the contrastive loss with the hyperbolic distance:

$$\mathcal{L}_{\text{hyp}}^{\text{self}} = -\sum_{i \in I} \log \frac{\exp(-d_{\mathbb{D}}(z_i, z_{j(i)})/\tau)}{\sum_{a \in A(i)} \exp(-d_{\mathbb{D}}(z_i, z_a)/\tau)} \quad (57)$$

but unlike Ge et al. (2023), they only have a hyperbolic branch and do not retain a Euclidean branch. Yue et al. (2023) also propose to extend the supervised contrastive learning method SupCon (Khosla et al., 2020) in the same way. Yue et al. (2023) also propose to train an adversarially robust contrastive learner that extends the Robust Contrastive Learning (RoCL) (Kim et al., 2020) method to hyperbolic space by replacing the Euclidean contrastive losses in RoCL's adversarial training loss with their hyperbolic contrastive loss:

$$\mathcal{L}_{\text{hyp}}^{\text{self}}(\tilde{x}, \{\tilde{x}^+, \tilde{x}^{\text{adv}}\}, \{\tilde{x}^-\}) + \lambda \mathcal{L}_{\text{hyp}}^{\text{self}}(\tilde{x}^{\text{adv}}, \tilde{x}^+, \{\tilde{x}^-\}) \quad (58)$$

where $\tilde{x}$ is a given image, $\tilde{x}^+$ is a positive example, $\tilde{x}^-$ is a negative example, and $\tilde{x}^{\text{adv}}$ is an adversarial example that is within δ of $\tilde{x}$. As in Ge et al. (2023), Yan et al. (2021) find that hyperbolic contrastive learning generally achieves small gains over its Euclidean counterparts.

Doan et al. (2023) tackles the Open World Object Detection (OWOD) task, by leveraging the object *unknownness* level with respect to the context. To this end, Doan et al. (2023) propose Hyp-OW consisting of three main parts: *Hyperbolic contrastive learning*, to learn a hierarchical class representation, *Super class regularizer*, to push semantically similar classes close, and *Adaptive relabeling*, to detect unknown objects using hyperbolic distance based relabeling. The hyperbolic contrastive loss is the usual contrastive loss with temperature (e.g., Eq. 57) performed on hyperbolic features, which are extracted from a Euclidean feature extractor and embedded into hyperbolic space using the exponential map. Positive and negative examples are drawn from both the batch $\mathcal{B}$ as well as a buffer $\mathcal{M}$. In super class regularization, a category p consisting of classes $S_p = c_1, \ldots, c_n$ is embedded as the hyperbolic average (using the hyperbolic average of Khrulkov et al. (2020)) of the hyperbolic embeddings of its constituent classes. Category embeddings are then supervised by the same contrastive loss at the category level. Finally, in adaptive relabeling, the maximum distance $\delta_{\mathcal{B}}$ from each matched (that is, has a groundtruth label, denoted $\mathbf{z}^m$) to every class centroid $\underline{\mathbf{z}_c}$ (where the class

centroid is computed by the hyperbolic average above):

$$\delta_{\mathcal{B}} = \max_{m \in \mathcal{B}, c \in \mathcal{K}} d_{\mathbb{D}}(z^m, \underline{\mathbf{z}_c}) \quad (59)$$

Unlabelled examples $\mathbf{z}^u$ are then labelled if they satisfy the condition

$$\min_{c \in \mathcal{K}} d_{\mathbb{D}}(\mathbf{z}^u, \underline{\mathbf{z}_c}) \leq \delta_{\mathcal{B}} \quad (60)$$

which essentially says that if an unlabelled example is "as certain" as some labelled example, it should be labelled.

Durrant & Leontidis (2023) also propose a hyperbolic self-supervised approach, using Ideal prototypes to extend masked Siamese networks Assran et al. (2022) to hyperbolic space. To do this, the dot product similarity used by Assran et al. (2022) is replaced with distance on the Poincaré ball. Similarities with prototypes are replaced with the Busemann function on the Poincaré ball. Finally, a hyperbolic projection head is used in place of an Euclidean projection head, using the hyperbolic linear layers of Shimizu et al. (2021).

Doan et al. (2023) tackles the Open World Object Detection (OWOD) task, by leveraging the object *unknownness* level with respect to the context. To this end, they propose Hyp-OW consisting of three main parts: *Hyperbolic contrastive learning*, to learn a hierarchical class representation, *Super class regularizer*, to push semantically similar classes close, and *Adaptive relabeling*, to detect unknown objects using hyperbolic distance based relabeling. Durrant & Leontidis (2023) also propose a hyperbolic self-supervised approach, where ideal prototypes are used to extend masked siamese networks Assran et al. (2022) to hyperbolic space.

5 Conclusions and Future Outlook

This survey provides an overview of the current state of affairs in hyperbolic deep learning for computer vision. Based on the organization of supervised and unsupervised literature, we conclude the survey by discussing which types of problems currently benefit most from hyperbolic learning and discussing open problems for future research.

5.1 When is Hyperbolic Learning Most Effective?

From current works, we identify four main axes of improvement that have come with the recent shift towards learning in hyperbolic space for computer vision:

- *Hierarchical learning.* The inherent links between hierarchical data and hyperbolic embeddings are well known. It is therefore not all too surprising to see that a wide range of works have used hyperbolic learning to improve

hierarchical objectives in computer vision. The ability to incorporate hierarchical knowledge, for example through hyperbolic embeddings or hierarchical hyperbolic logistic regression, has been utilized for several problems. Hierarchical learning in hyperbolic space can among others reduce error severity, resulting in smaller mistakes and more consistent retrieval, see e.g., Long et al. (2020), Dhall et al. (2020) and Yu et al. (2022b). This is a key property for example in medical domains, where large mistakes need to be avoided at all costs.

Hierarchical learning has also shown to have applications in zero-shot learning. By embedding class hierarchies in hyperbolic space and mapping examples of seen classes to their corresponding embedding, it becomes possible to generalize to examples of unseen classes (Liu et al., 2020). In general, hierarchical information between classes helps to structure the semantics of the task at hand, and embedding such knowledge in hyperbolic space is preferred over Euclidean space.

- *Few-sample learning.* Few-shot learning is popular in hyperbolic deep learning for computer vision. Many works have shown that consistent improvements can be made by performing this task with hyperbolic embeddings and prototypes, both with [e.g., (Zhang et al., 2022)] and without [e.g., (Khrulkov et al., 2020)] hierarchical knowledge. In few-shot learning, samples are scarce when it comes to generalization, and working in hyperbolic space consistently improves accuracy. These results indicate that hyperbolic space can generalize from fewer examples, with potential in domains where examples are scarce. This is already visible in the unsupervised domain, where generative learning is better in hyperbolic space when working with constrained data sources.

- *Robust learning.* Across several axes, hyperbolic learning has shown to be more robust. For example, hyperbolic embeddings improve out-of-distribution detection, provide a natural way to quantify uncertainty about samples [see e.g., (Ghadimi Atigh et al., 2022)], pinpoint unsupervised out-of-context samples [see e.g., (Ge et al., 2023)], and can improve robustness to adversarial attacks [see e.g., (Guo et al., 2022)]. Robustness and uncertainty are key challenges in deep learning in general, hyperbolic deep learning can provide a natural solution to robustify networks.

- *Low-dimensional learning.* For a lot of applications, networks, and embedding spaces need to be constrained, for example when learning on embedded devices or when visualizing data. In the unsupervised domain, hyperbolic learning consistently improves over Euclidean learning when working with smaller embedding spaces [see e.g., (Nagano et al., 2019)]. Similarly, the embedding space in supervised problems can be substantially reduced while maintaining downstream performance in hyper-

bolic space [see e.g., (Ghadimi Atigh et al., 2021)]. As such, hyperbolic learning has the potential to enable learning in compressed and embedded domains.

5.2 Open Research Questions

Hyperbolic learning has made an impact on computer vision with many promising avenues ahead. The field is however still in the early stages with many challenges and opportunities ahead. Three directions stand out:

- *Fully hyperbolic learning* Hyperbolic learning papers in computer vision commonly share one perspective: hyperbolic learning should be done in the embedding space. For the most part, the representation learning of earlier layers is done in Euclidean space, resulting in hybrid networks. Works from neuroscience indicate that for the earlier layers in neural networks, hyperbolic space can also play a prominent role (Chossat, 2020). Recently, Zhang et al. (2023) have shown that spatial relations in the hippocampus are more hyperbolic than Euclidean.
  Learning deep networks fully in hyperbolic space requires rethinking all layers, from convolutions to self-attention and normalization. At the time of writing the survey, two works have made steps in this direction. Bdeir et al. (2023) introduce a hyperbolic convolutional network in the Lorentz model of hyperbolic space. They outline how to perform convolutions, batch normalization, and residual connections. Simultaneously, van Spengler et al. (2023a) introduce Poincaré ResNet, with convolutions, residuals, batch normalization, and better network initialization in the Poincaré ball model. The works provide a foundation towards fully hyperbolic learning, but many open questions remain. Which model is most suitable for fully hyperbolic learning? Or do different layers work best in different models? And how can fully hyperbolic learning scale to ImageNet and beyond? Should each stage of the network have the same curvature? And how effective can hyperbolic networks become across all possible tasks compared to Euclidean networks? A lot more research is needed to answer these questions.

- *Computational challenges* Performing gradient-based learning in hyperbolic space changes how networks are optimized and how parameters behave. Compared to their Euclidean counterpart however, hyperbolic networks and embeddings can be numerically more unstable, with issues at the boundary of the ball (Moreira et al. 2023), vanishing gradients, and more. Moreover, hyperbolic operations can be more involved and computationally heavy depending on the used model, leading to less efficient networks. Such computational challenges are relevant for all domains of hyperbolic learning and a broader topic that is receiving attention.

- *Open source community* Modern deep learning libraries are centered around Euclidean geometry. Any new researcher in hyperbolic learning, therefore, does not have the opportunity to quickly implement networks and layers to get an intuition into its workings. Moreover, any new advances have to be either implemented from scratch or imported from code repositories of other papers. What is missing is an open-source community and a shared repository that houses advances in hyperbolic learning for computer vision. Such a community and code base is vital to get further traction and attract a wide audience, including practitioners. Whether it be part of existing libraries or as a separate library, continued development of open-source hyperbolic learning code is key for the future of the field. In recent years, several libraries have initiated learning and optimizing in hyperbolic space, including geoopt (Kochurov et al., 2020), geomstats (Miolane et al., 2020), manifolds.jl (Axen et al., 2021), and HypLL (van Spengler et al., 2023b). These libraries will form a great basis towards the development of hyperbolic learning.
- *Large and multimodal learning* In computer vision, and Artificial Intelligence in general, there is a strong trend towards learning at large scale and learning with multiple modalities, e.g., image-text or video-audio models. It is therefore a natural desire for the field to arrive at hyperbolic foundation models. While early work has shown that large-scale and/or multimodal learning is viable with hyperbolic embeddings (Desai et al., 2023), hyperbolic foundation models form a longer-term commitment as they require solutions to all open problems mentioned above, from stable, fully hyperbolic learning to continued open source development.
- *Multiple hyperbolic models* Unique to hyperbolic geometry is the existence of multiple models to perform numerical operations. Multiple papers have shown that different operations are preferred in different models. For example, for computing the mean of a distribution, the Klein model (Dai et al., 2021) is preferred over the Poincaré ball model as this avoids having to compute the expensive Fréchet mean (van Spengler et al., 2023a; Khrulkov et al., 2020). For representation layers, multiple papers advocate for the Lorentz model over the Poincaré ball model as it is faster and more robust (Chen et al., 2021; Dai et al., 2021). Recently Mishne et al. (2023) have also investigated the limitations of the Poincaré ball model and the Lorentz model. As shown in Mishne et al. (2023)), while the Poincaré ball model has a larger capacity to accurately represent points, the Lorentz model is stronger in optimization and training perspectives. It remains an open question which model is most suitable overall and whether one model suits all or we should employ different hyperbolic models for different operations.

References

Ahmad, O., & Lecue, F. (2022). Fisheyehdk: Hyperbolic deformable kernel learning for ultra-wide field-of-view image recognition. In *AAAI conference on artificial intelligence*.

Amin, F., Mondal, A., & Mathew, J. (2022). Deep semantic hashing with structure-semantic disagreement correction via hyperbolic metric learning. In *International workshop on multimedia signal processing*.

Anvekar, T., & Bazazian, D. (2023). Gpr-net: Geometric prototypical network for point cloud few-shot learning. arXiv.

Araño, K. A., Orsenigo, C., Soto, M., & Vercellis, C. (2021). Multimodal sentiment and emotion recognition in hyperbolic space. Expert Systems with Applications.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*.

Asano, Y. M., Rupprecht, C., & Vedaldi, A. (2019). Self-labelling via simultaneous clustering and representation learning. arXiv preprint arXiv:1911.05371.

Assran, M., Caron, M., Misra, I., Bojanowski, P., Bordes, F., Vincent, P., Joulin, A., Rabbat, M., & Ballas, N. (2022). Masked siamese networks for label-efficient learning. In *European conference on computer vision*. Berlin: Springer (pp 456–473).

Axen, S. D., Baran, M., Bergmann, & R., Rzecki, K. (2021). Manifolds. JL: An extensible Julia framework for data analysis on manifolds. arXiv preprint arXiv:2106.08777.

Bachmann, G., Bécigneul, G., & Ganea, O. (2020). Constant curvature graph convolutional networks. In *International conference on machine learning*.

Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J. S., Freymann, J. B., Farahani, K., & Davatzikos, C. (2017). Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Scientific data.

Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R. T., Berger, C., Ha, S. M., Rozycki, M., et al. (2018). Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. arXiv.

Balazevic, I., Allen, C., & Hospedales, T. (2019). Multi-relational poincaré graph embeddings. In *Advances in neural information processing systems*.

Bdeir, A., Schwethelm, K., & Landwehr, N. (2023). Hyperbolic geometry in computer vision: A novel framework for convolutional neural networks. arXiv.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. arXiv.

Bose, J., Smofsky, A., Liao, R., Panangaden, P., & Hamilton, W. (2020). Latent variable modelling with hyperbolic normalizing flows. In *International conference on machine learning*.

Bridson, M. R., & Haefliger, A. (2013). *Metric Spaces of Non-positive Curvature* (Vol. 319). Berlin: Springer.

Cannon, J. W., Floyd, W. J., Kenyon, R., Parry, W. R., et al. (1997). Hyperbolic geometry. Flavors of. *Geometry, 31*, 59–115.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *international conference on computer vision*.

Cetin, E., Chamberlain, B., Bronstein, M., & Hunt, J. J. (2022). Hyperbolic deep reinforcement learning. arXiv preprint arXiv:2210.01542.

Chamberlain, B. P., Hardwick, S. R., Wardrope, D. R., Dzogang, F., Daolio, F., & Vargas, S. (2019). Scalable hyperbolic recommender systems. arXiv preprint arXiv:1902.08648.

Chami, I., Ying, Z., Ré C, & Leskovec, J. (2019). Hyperbolic graph convolutional neural networks. In *Advances in neural information processing systems*.

Chami, I., Gu, A., Chatziafratis, V., & Ré, C. (2020a). From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. In *Advances in neural information processing systems*.

Chami, I., Wolf, A., Juan, D. C., Sala, F., Ravi, S., & Ré, C. (2020b). Low-dimensional hyperbolic knowledge graph embeddings. arXiv.

Chen, B., Peng, W., Cao, X., & Röning, J. (2022). Hyperbolic uncertainty aware semantic segmentation. *Transactions on Intelligent Transportation Systems*.

Chen, G., Qiao, L., Shi, Y., Peng, P., Li, J., Huang, T., Pu, S., & Tian, Y. (2020a). Learning open set network with discriminative reciprocal points. In *European conference on computer vision*.

Chen, J., Qin, J., Shen, Y., Liu, L., Zhu, F., & Shao, L. (2020b). Learning attentive and hierarchical representations for 3d shape recognition. In *European conference on computer vision*.

Chen, J., Jin, Z., Wang, Q., & Meng, H. (2023). Self-supervised 3D behavior representation learning based on homotopic hyperbolic embedding. *IEEE Transactions on Image Processing, 32*, 6061–6074.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020c). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*.

Chen, W., Han, X., Lin, Y., Zhao, H., Liu, Z., Li, P., Sun, M., & Zhou, J. (2021). Fully hyperbolic neural networks. arXiv.

Cho, H., DeMeo, B., Peng, J., & Berger, B. (2019). Large-margin classification in hyperbolic space. In *International conference on artificial intelligence and statistics*.

Cho, S., Lee, J., Park, J., & Kim, D. (2022). A rotated hyperbolic wrapped normal distribution for hierarchical representation learning. arXiv.

Chossat, P. (2020). The hyperbolic model for edge and texture detection in the primary visual cortex. *The Journal of Mathematical Neuroscience*.

Choudhary, N., & Reddy, C. K. (2022). Towards scalable hyperbolic neural networks using Taylor series approximations. arXiv.

Cui, Y., Yu, Z., Peng, W., & Liu, L. (2022). Rethinking few-shot class-incremental learning with open-set hypothesis in hyperbolic geometry. arXiv.

Dai, J., Wu, Y., Gao, Z., & Jia, Y. (2021). A hyperbolic-to-hyperbolic graph convolutional network. In *Computer vision and pattern recognition*.

Dai, S., Gan, Z., Cheng, Y., Tao, C., Carin, L., & Liu, J. (2020). APO-VAE: Text generation in hyperbolic space. arXiv preprint arXiv:2005.00054.

Dasgupta, S. (2016). A cost function for similarity-based hierarchical clustering. In *ACM symposium on theory of computing*.

Dengxiong, X., & Kong, Y. (2023). Ancestor search: Generalized open set recognition via hyperbolic side information learning. In *Winter conference on applications of computer vision*.

Desai, K., Nickel, M., Rajpurohit, T., Johnson, J., & Vedantam, R. (2023). Hyperbolic image-text representations. arXiv.

Dhall, A., Makarova, A., Ganea, O., Pavllo, D., Greeff, M., & Krause, A. (2020). Hierarchical image classification using entailment cone embeddings. In *Computer vision and pattern recognition workshops*.

Dhingra, B., Shallue, C. J., Norouzi, M., Dai, A. M., & Dahl, G. E. (2018). Embedding text in hyperbolic spaces. In *Workshop on graph-based methods for natural language processing*.

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using real nvp. arXiv.

Doan, T., Li, X., Behpour, S., He, W., Gou, L., & Ren, L. (2023). HYP-OW: Exploiting hierarchical structure learning with hyperbolic distance enhances open world object detection. arXiv preprint arXiv:2306.14291.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations*.

Durrant, A., & Leontidis, G. (2023). Hmsn: Hyperbolic self-supervised learning by clustering with ideal prototypes. arXiv preprint arXiv:2305.10926.

Ermolov, A., Mirvakhabova, L., Khrulkov, V., Sebe, N., & Oseledets, I. (2022). Hyperbolic vision transformers: Combining improvements in metric learning. In *Computer vision and pattern recognition*.

Fan, X., Yang, C. H., & Vemuri, B. C. (2022). Nested hyperbolic spaces for dimensionality reduction and hyperbolic NN design. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 356–365).

Fang, P., Harandi, M., & Petersson, L. (2021). Kernel methods in hyperbolic spaces. In *international conference on computer vision*.

Fang, P., Harandi, M., Lan, Z., & Petersson, L. (2023a). Poincaré kernels for hyperbolic representations. *International Journal of Computer Vision*, pp. 1–23.

Fang, P., Harandi, M., Le, T., & Phung, D. (2023b). Hyperbolic geometry in computer vision: A survey. arXiv preprint arXiv:2304.10764.

Franco, L., Mandica, P., Munjal, B., & Galasso, F. (2023). Hyperbolic self-paced learning for self-supervised skeleton-based action representations. In *International Conference on Learning Representations*.

Ganea, O., Bécigneul, G., & Hofmann, T. (2018a). Hyperbolic entailment cones for learning hierarchical embeddings. In *International conference on machine learning*.

Ganea, O., Bécigneul, G., & Hofmann, T. (2018b). Hyperbolic neural networks. In *Advances in neural information processing systems*.

Gao, Z., Wu, Y., Jia, Y., & Harandi, M. (2021). Curvature generation in curved spaces for few-shot learning. In *international conference on computer vision*.

Gao, Z., Wu, Y., Jia, Y., & Harandi, M. (2022). Hyperbolic feature augmentation via distribution estimation and infinite sampling on manifolds. In *Advances in neural information processing systems*.

Gao, Z., Xu, C., Li, F., Jia, Y., Harandi, M., & Wu, Y. (2023). Exploring data geometry for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 24325–24334).

Ge, S., Mishra, S., Kornblith, S., Li, C. L., & Jacobs, D. (2023). Hyperbolic contrastive learning for visual representations beyond

objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp 6840–6849).

Ghadimi Atigh, M., Keller-Ressel, M., & Mettes, P. (2021). Hyperbolic Busemann learning with ideal prototypes. In *Advances in neural information processing systems*.

Ghadimi Atigh, M., Schoep, J., Acar, E., van Noord, N., & Mettes, P. (2022). Hyperbolic image segmentation. In *Computer vision and pattern recognition*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. (2018). Recent advances in convolutional neural networks. *Pattern recognition*.

Gulcehre, C., Denil, M., Malinowski, M., Razavi, A., Pascanu, R., Hermann, K. M., Battaglia, P., Bapst, V., Raposo, D., Santoro, A., et al. (2019). Hyperbolic attention networks. In *International conference on learning representations*.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in neural information processing systems*.

Gulshad, S., Long, T., & van Noord, N. (2023). Hierarchical explanations for video action recognition. arXiv.

Guo, H., Tang, J., Zeng, W., Zhao, X., & Liu, L. (2021). Multi-modal entity alignment in hyperbolic space. *Neurocomputing*.

Guo, Y., Wang, X., Chen, Y., & Yu, S. X. (2022). Clipped hyperbolic classifiers are super-hyperbolic classifiers. In *Computer vision and pattern recognition*.

Gupta, A., Dollar, P., & Girshick, R. (2019). LVIS: A dataset for large vocabulary instance segmentation. In *Computer vision and pattern recognition*.

Hamann, M. (2018). On the tree-likeness of hyperbolic spaces. In *Mathematical proceedings of the Cambridge philosophical society*, Cambridge: Cambridge University Press (pp. 345–361).

Hamzaoui, M., Chapel, L., Pham, M. T., & Lefèvre, S. (2023). Hyperbolic prototypical network for few shot remote sensing scene classification. *Pattern Recognition Letters*.

Han, S., Cai, R., Cui, Y., Yu, Z., Hu, Y., & Kot, A. (2023). Hyperbolic face anti-spoofing. arXiv preprint arXiv:2308.09107.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Computer vision and pattern recognition*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter S. (2017). Gans trained by a two time-scale update rule converge to a local NASH equilibrium. In *Advances in neural information processing systems*.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). beta-VAE: Learning basic visual concepts with a constrained variational framework. In *international conference on learning representations*.

Hong, J., Fang, P., Li, W., Han, J., Petersson, L., & Harandi, M. (2023a). Curved geometric networks for visual anomaly recognition. *IEEE transactions on neural networks and learning systems*.

Hong, J., Hayder, Z., Han, J., Fang, P., Harandi, M., & Petersson, L. (2023b). Hyperbolic audio-visual zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 7873–7883).

Hsu, J., Gu, J., Wu, G., Chiu, W., & Yeung, S. (2021). Capturing implicit hierarchical structure in 3d biomedical images with self-supervised hyperbolic representations. In *Advances in neural information processing systems*.

Huang, W., Yu, Y., Xu, H., Su, Z., & Wu, Y. (2023). Hyperbolic music transformer for structured music generation. *IEEE Access, 11*, 26893–26905.

Iscen, A., Tolias, G., Avrithis, Y., Furon, T., & Chum, O. (2017). Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations. In *Computer vision and pattern recognition*.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Computer vision and pattern recognition*.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. In *Computer vision and pattern recognition*.

Kasarla, T., Burghouts, G., van Spengler, M., van der Pol, E., Cucchiara, R., & Mettes, P. (2022). Maximum class separation as inductive bias in one matrix. In *Advances in neural information processing systems*.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM Computing Surveys*.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. In *Advances in neural information processing systems*.

Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., & Lempitsky, V. (2020). Hyperbolic image embeddings. In *Computer vision and pattern recognition*.

Kim, M., Tack, J., & Hwang, S. J. (2020). Adversarial self-supervised contrastive learning. In *Advances in neural information processing systems*.

Kim, S., Jung, B., & Kwak, S. (2022). HIER: Metric learning beyond class labels via hierarchical regularization. arXiv.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv.

Klimovskaia, A., Lopez-Paz, D., Bottou, L., & Nickel, M. (2020). Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature communications*.

Kochurov, M., Karimov, R., & Kozlukov, S. (2020). Geoopt: Riemannian optimization in pytorch. arXiv preprint arXiv:2005.02819.

Law, M., Liao, R., Snell, J., & Zemel, R. (2019). Lorentzian distance learning for hyperbolic representations. In *International conference on machine learning* (pp. 3672–3681).

Lazcano, D., Franco, N. F., & Creixell, W. (2021). HGAN: Hyperbolic generative adversarial network. *IEEE Access*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*.

Leimeister, M., & Wilson, B. J. (2018). Skip-gram word embeddings in hyperbolic space. arXiv.

Leng, Z., Wu, S. C., Saleh, M., Montanaro, A., Yu, H., Wang, Y., Navab, N., Liang, X., & Tombari, F. (2023). Dynamic hyperbolic attention network for fine hand-object reconstruction. In *Proceedings of the IEEE/CVF international conference on computer Vision* (pp. 14894–14904).

Li, A., Yang, B., Hussain, F. K., & Huo, H. (2022). HSR: Hyperbolic social recommender. *Information Sciences, 585*, 275–288.

Li, H., Jiang, H., Ye, D., Wang, Q., Du, L., Zeng, Y., Wang, Y., Chen, C., et al. (2023a). Dhgat: Hyperbolic representation learning on dynamic graphs via attention networks. *Neurocomputing* p. 127038.

Li, L., Zhang, Y., & Wang, S. (2023b). The Euclidean space is evil: Hyperbolic attribute editing for few-shot image generation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 22714–22724).

Li, Y. L., Wu, X., Liu, X., Dou, Y., Ji, Y., Zhang, J., Li, Y., Tan, J., Lu, X., & Lu, C. (2023c). From isolated islands to Pangea: Unifying semantic space for human action understanding. arXiv.

Lin, F., Bai, B., Guo, Y., Chen, H., Ren, Y., & Xu, Z. (2023a). MHCN: A hyperbolic neural network model for multi-view hierarchical clustering. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16525–16535).

Lin, F., Yue, Y., Hou, S., Yu, X., Xu, Y., Yamada, K. D., & Zhang, Z. (2023b). Hyperbolic chamfer distance for point cloud comple-

tion. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 14595–14606).

Lin, Y. W. E., Coifman, R. R., Mishne, G., & Talmon, R. (2023c). Hyperbolic diffusion embedding and distance for hierarchical representation learning. arXiv preprint arXiv:2305.18962.

Liu, Q., Nickel, M., & Kiela, D. (2019). Hyperbolic graph neural networks. In *Advances in neural information processing systems*.

Liu, S., Chen, J., Pan, L., Ngo, C. W., Chua, T. S., & Jiang, Y. G. (2020). Hyperbolic visual embedding learning for zero-shot recognition. In *Computer vision and pattern recognition*.

Long, T., & van Noord, N. (2023). Cross-modal scalable hyperbolic hierarchical clustering. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16655–16664).

Long, T., Mettes, P., Shen, H. T., & Snoek, C. G. M. (2020). Searching for actions on the hyperbole. In *Computer vision and pattern recognition*.

Lou, A., Katsman, I., Jiang, Q., Belongie, S., Lim, S. N., & De Sa, C. (2020). Differentiating through the fréchet mean. In *International conference on machine learning*.

Ma, R., Fang, P., Drummond, T., & Harandi, M. (2022). Adaptive poincaré point to set distance for few-shot classification. In *AAAI conference on artificial intelligence*.

Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., & The, Y. W. (2019). Continuous hierarchical representations with poincaré variational auto-encoders. In *Advances in neural information processing systems*, (vol. 32).

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al. (2014). The multimodal brain tumor image segmentation benchmark (BRATS). In: *IEEE transactions on medical imaging*.

Mettes, P., Van der Pol, E., & Snoek, C. (2019). Hyperspherical prototype networks. In *Advances in neural information processing systems*.

Miolane, N., Guigui, N., Le Brigant, A., Mathe, J., Hou, B., Thanwerdas, Y., Heyder, S., Peltre, O., Koep, N., Zaatiti, H., et al. (2020). Geomstats: A python package for riemannian geometry in machine learning. *The Journal of Machine Learning Research, 21*(1), 9203–9211.

Mirvakhabova, L., Frolov, E., Khrulkov, V., Oseledets, I., & Tuzhilin, A. (2020). Performance of hyperbolic geometry models on top-n recommendation tasks. In *ACM conference on recommender systems*.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv.

Mishne, G., Wan, Z., Wang, Y., & Yang, S. (2023). The numerical stability of hyperbolic representation learning. In *International conference on machine learning, PMLR* (pp. 24925–24949).

Monath, N., Zaheer, M., Silva, D., McCallum, A., & Ahmed, A. (2019). Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *International conference on knowledge discovery & data mining*.

Montanaro, A., Valsesia, D., & Magli, E. (2022). Rethinking the compositionality of point clouds through regularization in the hyperbolic space. arXiv.

Moreira, G., Marques, M., Costeira, J. P., & Hauptmann, A. (2023). Hyperbolic vs Euclidean embeddings in few-shot learning: Two sides of the same coin. arXiv preprint arXiv:2309.10013.

Nagano, Y., Yamaguchi, S., Fujita, Y., & Koyama, M. (2019). A wrapped normal distribution on hyperbolic space for gradient-based learning. In *International conference on machine learning*.

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, (vol. 30).

Nickel, M., & Kiela, D. (2018). Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *International conference on machine learning*.

Noy, N. F., & Hafner, C. D. (1997). The state of the art in ontology design: A survey and comparative review. AI Magazine.

Onghena, P., Gigli, L., & Velasco-Forero, S. (2023). Rotation-invariant hierarchical segmentation on poincare ball for 3d point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1765–1774).

Park, J. H., Choe, J., Bae, I., & Jeon, H. G. (2023). Learning affinity with hyperbolic representation for spatial propagation. In *International conference on machine learning*.

Peng, W., Varanka, T., Mostafa, A., Shi, H., & Zhao G. (2021). Hyperbolic deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Qu, E., & Zou, D. (2022a). Autoencoding hyperbolic representation for adversarial generation. arXiv.

Qu, E., & Zou, D. (2022b). Hyperbolic neural networks for molecular generation. arXiv preprint arXiv:2201.12825.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *international conference on machine learning*.

Ratcliffe, J. G. (1994). *Foundations of hyperbolic manifolds* (Vol. 149). Berlin: Springer.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic back-propagation and approximate inference in deep generative models. In *International conference on machine learning*.

Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., & Cohen-Or, D. (2021). Encoding in style: a stylegan encoder for image-to-image translation. In *Computer vision and pattern recognition*.

Sala, F., De Sa, C., Gu, A., & Ré, C. (2018). Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems*.

Sarkar, R. (2011). Low distortion Delaunay embedding of trees in hyperbolic plane. *International symposium on graph drawing* (pp. 355–366). Berlin: Springer.

Schuler, K. K. (2005). *VerbNet: A broad-coverage, comprehensive verb lexicon*. Philadelphia: University of Pennsylvania.

Shimizu, R., Mukuta, Y., & Harada, T. (2021). Hyperbolic neural networks++. In *International conference on learning representations*.

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In *Advances in neural information processing systems*.

Sonthalia, R., & Gilbert, A. (2020). Tree! I am no tree! I am a low dimensional hyperbolic embedding. *Advances in Neural Information Processing Systems, 33*, 845–856.

van Spengler, M., Berkhout, E., & Mettes, P. (2023a). Poincaré resnet. arXiv.

van Spengler, M., Wirth, P., & Mettes, P. (2023b). Hypll: The hyperbolic learning library. arXiv preprint arXiv:2306.06154.

Sun, J., Cheng, Z., Zuberi, S., Pérez, F., & Volkovs, M. (2021). HGCF: Hyperbolic graph convolution networks for collaborative filtering. *Proceedings of the web conference, 2021* (pp. 593–601).

Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al. (2021b). Sparse r-CNN: End-to-end object detection with learnable proposals. In *Computer vision and pattern recognition*.

Surís, D., Liu, R., & Vondrick, C. (2021). Learning the predictability of the future. In *Computer vision and pattern recognition*.

Tifrea, A., Bécigneul, G., & Ganea, O. E. (2019). Poincar'e glove: Hyperbolic word embeddings. In *International conference on learning representations*.

Tong, J., Yang, F., Yang, S., Dong, E., Du, S., Wang, X., & Yi, X. (2022). Hyperbolic cosine transformer for lidar 3d object detection. arXiv.

Trpin, A., & Boshkoska, B. (2022). Face recognition with a hyperbolic metric classification model. In *International convention on information, communication and electronic technology*.

Ungar, A. A. (2005). Gyrovector spaces and their differential geometry. *Nonlinear Functional Analysis and Applications, 10*(5), 791–834.

Ungar, A. A. (2008). A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics, 1*(1), 1–194.

Ungar, A. A. (2012). *Beyond the Einstein addition law and its gyroscopic Thomas precession: The theory of gyrogroups and gyrovector spaces* (Vol. 117). Berlin: Springer.

Valada, A. (2022). On hyperbolic embeddings in object detection. In *German conference on pattern recognition*.

Verbeek, K., & Suri, S. (2014). Metric embedding, hyperbolic space, and social networks. In: *Proceedings of the thirtieth annual symposium on Computational geometry* (pp. 501–510).

Vinh, T. D. Q., Tay, Y., Zhang, S., Cong, G., & Li, X. L. (2018). Hyperbolic recommender systems. arXiv preprint arXiv:1809.01703.

Vinh Tran, L., Tay, Y., Zhang, S., Cong, G., & Li, X. (2020). Hyperml: A boosting metric learning approach in hyperbolic space for recommender systems. In *Proceedings of the 13th international conference on web search and data mining* (pp. 609–617).

Wang, D., & Wang, Y. (2018). An improved cost function for hierarchical cluster trees. arXiv.

Wang, L., Hu, F., Wu, S., & Wang, L. (2021). Fully hyperbolic graph convolution network for recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*.

Wang, S., Kang, Q., She, R., Wang, W., Zhao, K., Song, Y., & Tay, W. P. (2023a). Hypliloc: Towards effective lidar pose regression with hyperbolic fusion. arXiv.

Wang, Y., Wang, H., Lu, W., & Yan, Y. (2023). Hygge: Hyperbolic graph attention network for reasoning over knowledge graphs. *Information Sciences, 630*, 190–205.

Weng, Z., Ogut, M. G., Limonchik, S., & Yeung, S. (2021). Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision. In *Computer vision and pattern recognition* (pp. 2603–2612).

Wu, Z., Jiang, D., Hsieh, C. Y., Chen, G., Liao, B., Cao, D., & Hou T. (2021). Hyperbolic relational graph convolution networks plus: a simple but highly efficient QSAR-modeling method. *Briefings in Bioinformatics*.

Xu, Y., Mu, L., Ji, Z., Liu, X., & Han, J. (2022). Meta hyperbolic networks for zero-shot learning. *Neurocomputing*.

Yan, J., Luo, L., Deng, C., & Huang, H. (2021). Unsupervised hyperbolic metric learning. In *Computer Vision and Pattern Recognition*.

Yan, J., Luo, L., Deng, C., & Huang, H. (2023). Adaptive hierarchical similarity metric learning with noisy labels. *IEEE Transactions on Image Processing*.

Yang, M., Zhou, M., Li, Z., Liu, J., Pan, L., Xiong, H., & King, I. (2022a). Hyperbolic graph neural networks: A review of methods and applications. arXiv preprint arXiv:2202.13852.

Yang, M., Zhou, M., Liu, J., Lian, D., & King, I. (2022b). HRCF: Enhancing collaborative filtering via hyperbolic geometric regularization. In *Proceedings of the ACM web conference*.

Yang, M., Zhou, M., Ying, R., Chen, Y., & King, I. (2023). Hyperbolic representation learning: Revisiting and advancing. arXiv preprint arXiv:2306.09118

Yu, K., Visweswaran, S., & Batmanghelich, K. (2020). Semi-supervised hierarchical drug embedding in hyperbolic space. *Journal of Chemical Information and Modeling*.

Yu, Z., Nguyen, T., Gal, Y., Ju, L., Chandra, S. S., Zhang, L., Bonnington, P., Mar, V., Wang, Z., & Ge, Z. (2022b). Skin lesion recognition with class-hierarchy regularized hyperbolic embeddings. In *International conference on medical image computing and computer-assisted intervention*.

Yue, Y., Lin, F., Yamada, K. D., & Zhang, Z. (2023). Hyperbolic contrastive learning. arXiv.

Zhang, B., Jiang, H., Feng, S., Li, X., Ye, Y., & Ye, R. (2022). Hyperbolic knowledge transfer with class hierarchy for few-shot learning. In *International joint conference on artificial intelligence*.

Zhang, H., Rich, P. D., Lee, A. K., & Sharpee, T. O. (2023). Hippocampal spatial representations exhibit a hyperbolic geometry that expands with experience. *Nature Neuroscience*.

Zhang, Y., Luo, L., Xian, W., & Huang, H. (2021a). Learning better visual data similarities via new grouplet non-euclidean embedding. In *International conference on computer vision*.

Zhang, Y., Wang, X., Shi, C., Jiang, X., Ye Y. (2021b). Hyperbolic graph attention network. IEEE Transactions on Big Data.

Zhang, Y., Wang, X., Shi, C., Liu, N., & Song, G. (2021c). Lorentzian graph convolutional networks. In *Proceedings of the web conference*.

Zhu, Y., Zhou, D., Xiao, J., Jiang, X., Chen, X., & Liu, Q. (2020). Hypertext: Endowing fasttext with hyperbolic geometry. In *Empirical methods in natural language processing*