Universität Basel

HSLU Hochschule Luzern

Hyperbolic Learning in Action

Non-Euclidean Geometry

Lionetti Simone

Gonzalez-Jimenez Alvaro

Friday 14th February, 2025

Hyperbolic Learning in Action

Website and material:

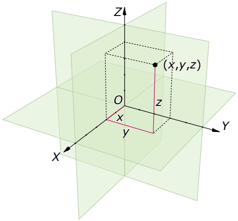
`https://digital-dermatology.github.io/hyperbolic-learning-tutorial`

Motivation

Euclidean geometry

Humans perceive the world as three-dimensional Euclidean space.

Width, height, and depth are natural concepts.

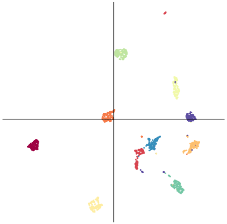


Jorge Stolfi, Public domain, via Wikimedia commons

Computer linear algebra assumes Euclidean space.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

Most Machine Learning is based on Euclidean space.



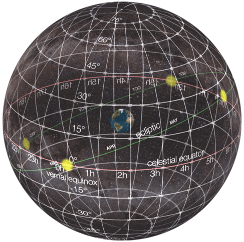
Spherical geometry

Some problems are naturally treated on the sphere.

Earth surface

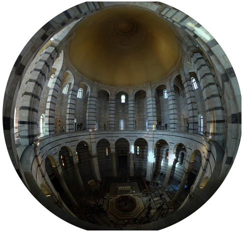
U.S. Government, Public domain,
via Wikimedia commons

Celestial sphere



ChristianReady, CC BY-SA 4.0,
via Wikimedia commons

Fisheye camera



Spike, CC BY-SA 4.0,
via Wikimedia commons

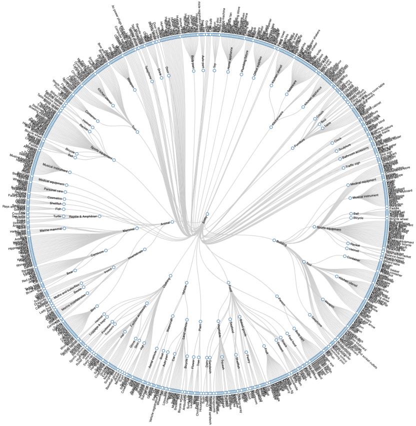
More subtly, cosine distance is often used in embedding spaces.

Hyperbolic geometry

Hyperbolic geometry
is less common in nature...

Toby Hudson, CC BY-SA 3.0,
via Wikimedia commons

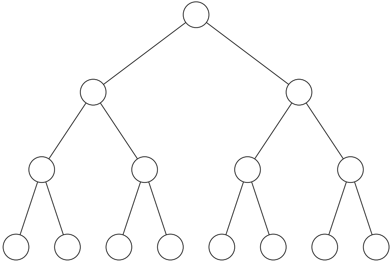
...but common in data!



[Schumann et al 2021]
CC BY-SA 4.0

Hierarchies

Tree structures with splitting at each level:



The number of leaves grows *exponentially* with the number of levels.

This is often the structure of:

- Classification categories
- Images and their parts
- Words and their relations
- Tree graphs, by definition
- ...

Ubiquitous in Machine Learning!

Program for 45 minutes

1. Motivation
2. Curvature
   - 2.1 Construction
   - 2.2 Properties
3. Hyperbolic geometry
   - 3.1 Lorentz hyperboloid model
   - 3.2 Poincaré ball model
   - 3.3 Isometries

Inspired by the tutorial on
Hyperbolic Representation Learning at ECCV 2022
by Mettes, Ghadimi Athig, Keller-Ressel, Gu, Yeung

Backgrounds

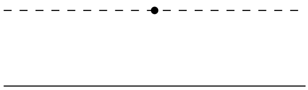
What is your background?

Curvature

Axioms of Euclidean geometry

Euclid's *Elements* (roughly 300 BC):

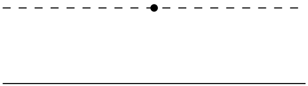
1. Unique line segment through two distinct points.
2. Line segments may be indefinitely extended.
3. Unique circle of given center and radius.
4. All right angles are congruent.
5. Given a line and a point not on it,
   there is a unique line through the given point
   that does not intersect the given line.



Axioms of Euclidean geometry

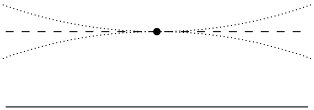
Euclid's *Elements* (roughly 300 BC):

1. Unique line segment through two distinct points.
2. Line segments may be indefinitely extended.
3. Unique circle of given center and radius.
4. All right angles are congruent.
5. Given a line and a point not on it,
   there is a unique line through the given point
   that does not intersect the given line.



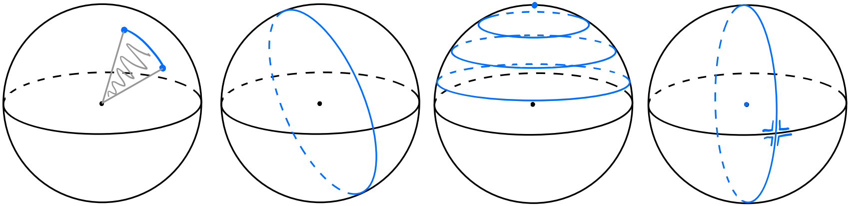
Concepts:

1. Geodesic segments
2. Geodesic lines
3. Distance
4. Orthogonality
5. (Flatness)



The sphere

1. The shortest path is in the plane with the two points and the center.
2. Great circles are geodesics, i.e. straight lines.
3. Same distance curves are "parallels".
4. Orthogonal plane great circles.
5. All great circles intersect in two points!



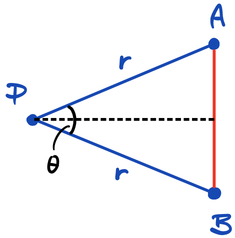
Euclidean distance between geodesics

An important characteristic of a geometry is the distance between geodesics crossing at a given angle θ, at a distance r from their intersection.

For the Euclidean plane

$$s_\theta = 2r\sin\frac{\theta}{2},$$

$$\left.\frac{\partial s_\theta}{\partial \theta}\right|_{\theta=0} = r.$$



Spherical distance between geodesics

For the sphere

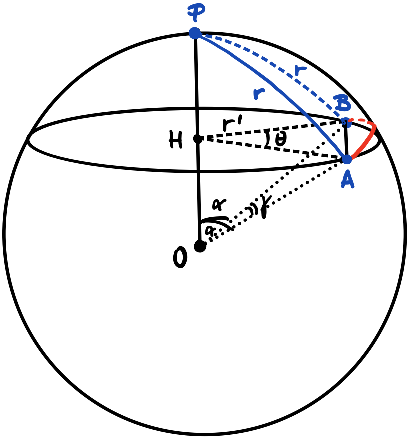
$$s_\theta = 2R \arcsin\left(\sin\frac{r}{R}\sin\frac{\theta}{2}\right),$$

$$\left.\frac{\partial s_\theta}{\partial\theta}\right|_{\theta=0} = R\sin\frac{r}{R}.$$

Note that for $R \to +\infty$

$$\left.\frac{\partial s_\theta}{\partial\theta}\right|_{\theta=0} \sim r,$$

the Euclidean case is recovered.



Surface element

Volume element

- Euclidean

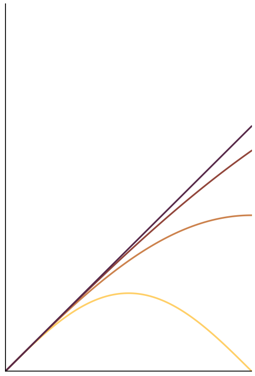
$$ds^2 = dr^2 + r^2 d\theta^2$$

- Spherical

$$ds^2 = dr^2 + R^2 \sin^2 \frac{r}{R} d\theta^2$$

Define:

$$ds^2 = dr^2 + [f(r)]^2 d\theta^2$$



Negative curvature, imaginary radius

Define curvature as

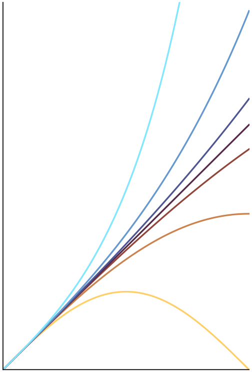
$$\kappa := R^{-2}.$$

$$f(r) = \frac{1}{\sqrt{\kappa}} \sin r\sqrt{\kappa}, \qquad \begin{cases} \text{Spherical} & \kappa > 0 \\ \text{Euclidean} & \kappa \to 0 \\ ? & \kappa < 0 \end{cases}.$$

Note $\kappa < 0$ means $R = i|R|$, rewrite

$$f(r) = -\frac{i}{\sqrt{-\kappa}} \sin\left(-ir\sqrt{-\kappa}\right) \quad i\sin(ix) = \sinh x$$

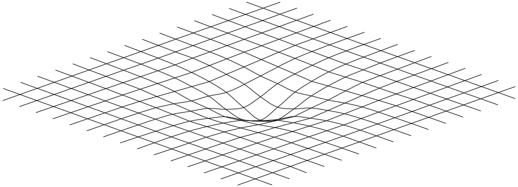
$$= \frac{1}{\sqrt{-\kappa}} \sinh\left(r\sqrt{-\kappa}\right)$$



The many faces of curvature

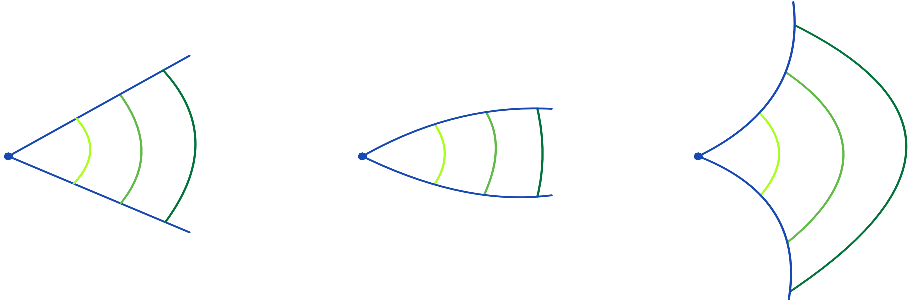
- **Intrinsic**: seen from within the space
  - Volume growth
  - Parallel postulate
  - Grid distortion
  - Parallel transport
  - Sum of internal angles
- **Extrinsic**: seen from a larger space
  - Principal curvatures
  - Gaussian curvature

- **Local**: at a given point in space
- **Global**: in a given region of space



Modified from Mysid, CC BY-SA 3.0,
via Wikimedia commons

Space growth



$\kappa = 0$

polynomial

$$S_{n-1} = \Omega_n r^{n-1}$$

$\kappa > 0$

bounded

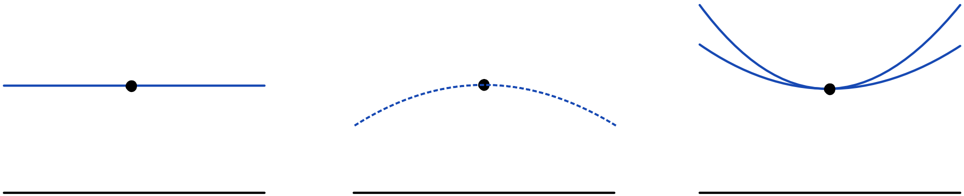
$$S^{+}_{n-1} = \Omega_n \left[R \sin \frac{r}{R} \right]^{n-1}$$

$\kappa < 0$

exponential

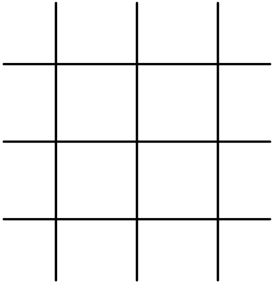
$$S^{-}_{n-1} = \Omega_n \left[|R| \sinh \frac{r}{|R|} \right]^{n-1}$$

Parallel postulate



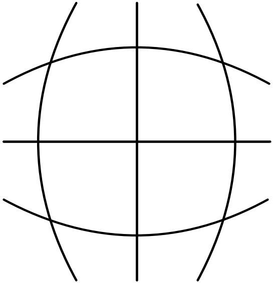
$\kappa = 0$ (one parallel) | $\kappa > 0$ (no parallel) | $\kappa < 0$ (many parallels)

Grid distortion



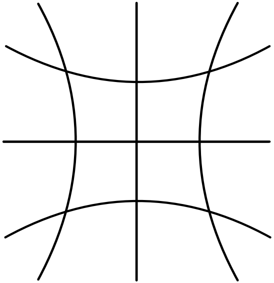
$\kappa = 0$

(flat)



$\kappa > 0$

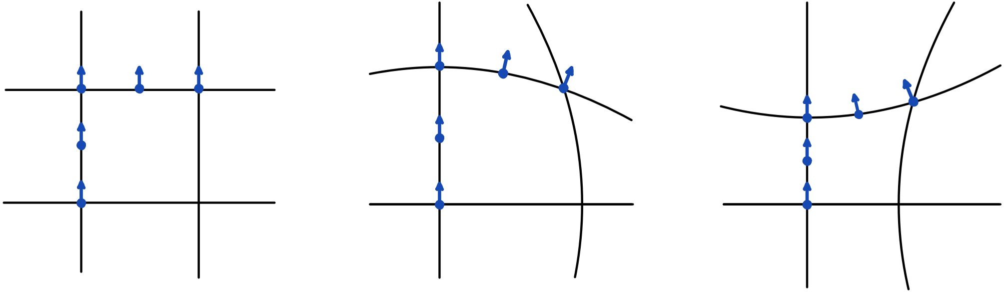
(barrel)



$\kappa < 0$

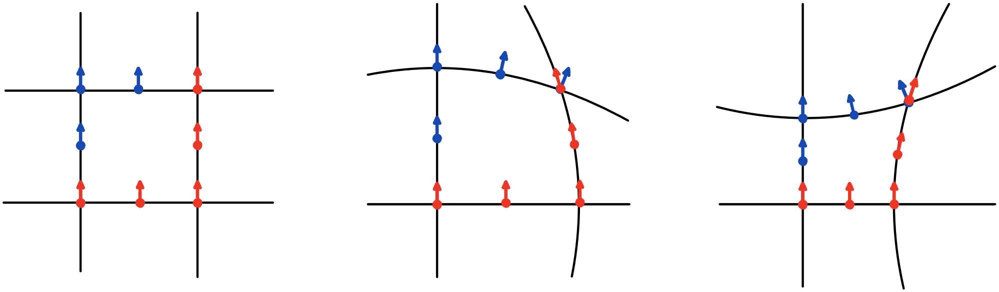
(pincushion)

Parallel transport



$\kappa = 0$ $\kappa > 0$ $\kappa < 0$

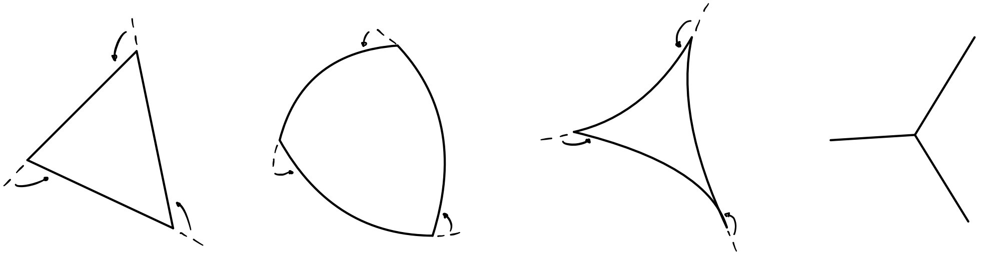
Parallel transport



$\kappa = 0$ $\kappa > 0$ $\kappa < 0$

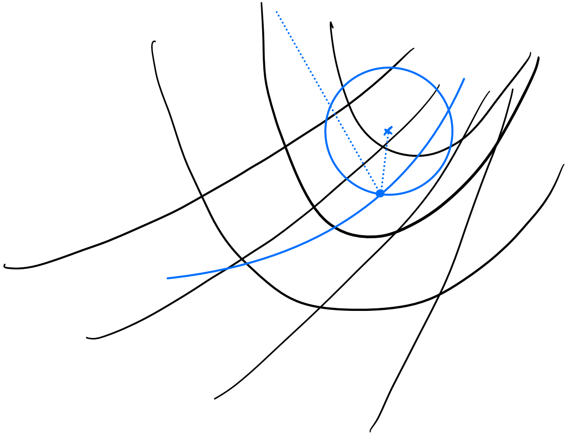
Triangles

Sum of internal angles



$\kappa = 0$	$\kappa > 0$	$\kappa < 0$	$\kappa \to -\infty$
$\sum_i \alpha_i = \pi$	$\sum_i \alpha_i > \pi$	$\sum_i \alpha_i < \pi$	$\sum_i \alpha_i = 0$

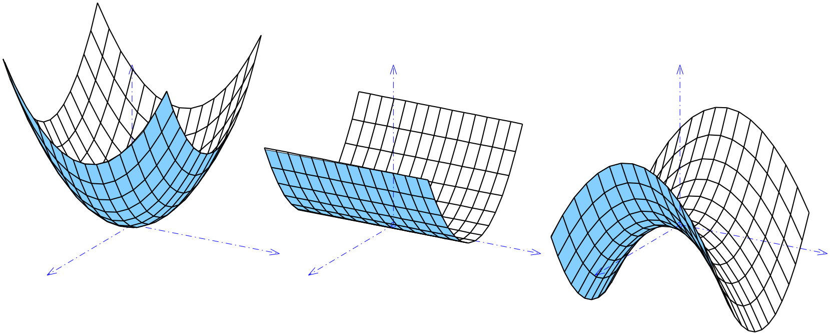
Principal curvatures



Principal curvatures are defined by the minimum and maximum radius.

Gaussian curvature

Gaussian curvature is the determinant of extrinsic curvatures,
it coincides with intrinsic curvature.



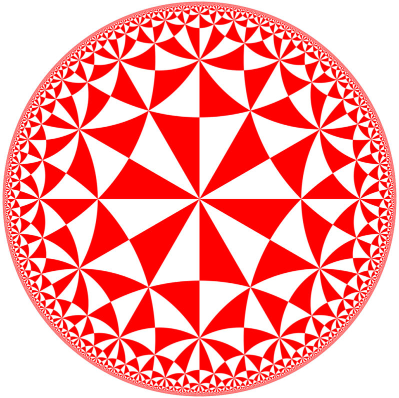
$\kappa > 0$ $\kappa = 0$ $\kappa < 0$

Hyperbolic space

Definition and history

Hyperbolic space is the space of **constant negative curvature**.

- Developed in the 19th century by Gauss, Lobachevsky, and Bolyai.
- Is the geometry of special relativity.
- Inspired art by Maurits C. Escher



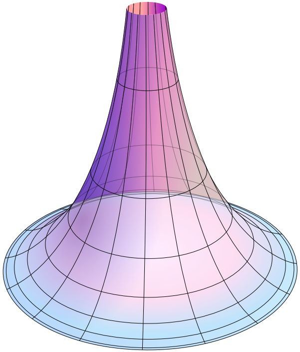
Hilbert's theorem

Bad piece of news:

There is no way to completely represent
the hyperbolic space of dimension n
in the Euclidean space of dimension $n+1$.

The best we can do is the tractoid.

This is why we have to resort to *models*.



Leonid 2, CC BY-SA 3.0,
via Wikimedia commons

Models of hyperbolic geometry

- Hyperboloid or Lorentz model
- Poincaré disc/ball
- Beltrami–Klein model
- Poincaré half-plane
- ...

All equivalent, but depending on the operation some may be more convenient.
A *conformal* model is one that preserves angles.

Minkowski space

Euclidean space $\mathbb{R}^n$ with an additional dimension

$$x = (x_0, x_1, \ldots, x_n) = (x_0, \vec{x})$$

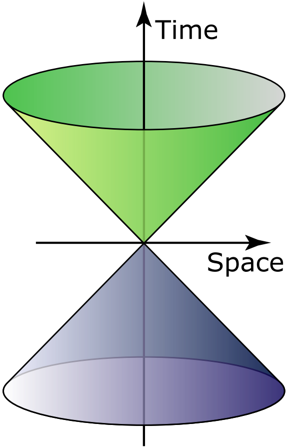
x_0 and $\vec{x}$ are called *time* and *space* components

Introduce the pseudo-scalar product

$$\begin{aligned}\langle x, y\rangle_{\mathcal{L}} &= x_0 y_0 - (x_1 y_1 + \cdots + x_n y_n)\\ &= x_0 y_0 - \vec{x} \cdot \vec{y}.\end{aligned}$$

This is not positive definite!

Example: $x^2 = 0$ when $x_0^2 = x_1^2 + x_2^2$



Lorentz hyperboloid model

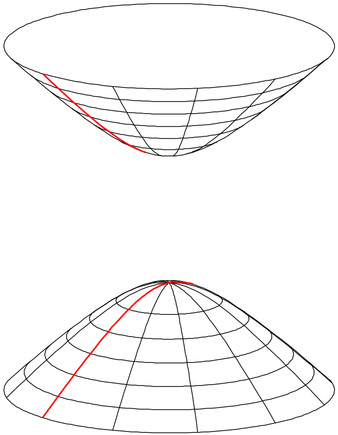
The Lorentz hyperboloid model is the manifold

$$x^2 = x_0^2 - \vec{x}^2 = -1/\kappa \quad \text{with} \quad x_0 > 0,$$

so x_0 is fully determined by $\vec{x}$

$$x_0 = \sqrt{\vec{x}^2 - 1/\kappa}.$$

An appropriate definition of distance is needed.



Ag2gaeh, CC BY-SA 4.0, via Wikimedia commons

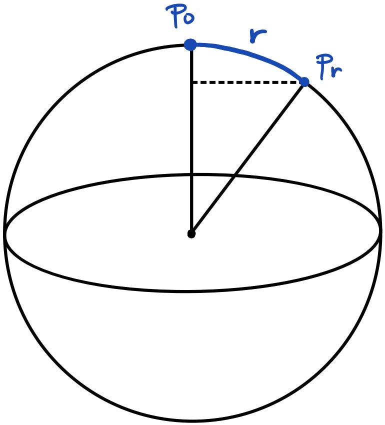
Distance in the sphere

For the n-dimensional sphere within $\mathbb{R}^{n+1}$, choose a meridian from the north pole

$$p_r = \begin{pmatrix} R\cos(r/R) \\ \hat{v}R\sin(r/R) \end{pmatrix}, \quad p_0 = \begin{pmatrix} R \\ \vec{0} \end{pmatrix}.$$

The distance r can be rewritten with the scalar product

$$\langle p_0, p_r \rangle = R^2 \cos\frac{r}{R},$$
$$r = R\arccos\frac{\langle p(0), p(r) \rangle}{R^2}.$$



Distance in the Lorentz hyperboloid

For the n-dimensional sphere within $\mathbb{R}^{n+1}$, choose a meridian from the north pole

$$p_r = \begin{pmatrix} R\cos(r/R) \\ \hat{v}R\sin(r/R) \end{pmatrix}, \quad p_0 = \begin{pmatrix} R \\ \vec{0} \end{pmatrix}.$$

The distance r can be rewritten with the scalar product

$$\langle p_0, p_r \rangle = R^2 \cos\frac{r}{R},$$
$$r = R \arccos\frac{\langle p(0), p(r) \rangle}{R^2}.$$

For the n-dimensional Lorentz hyperboloid in $(1, n)$ Minkowski space

$$p_r = \begin{pmatrix} \frac{1}{\sqrt{-\kappa}}\cosh\big(r\sqrt{-\kappa}\big) \\ \frac{\hat{v}}{\sqrt{-\kappa}}\sinh\big(r\sqrt{-\kappa}\big) \end{pmatrix}, \; p_0 = \begin{pmatrix} \frac{1}{\sqrt{-\kappa}} \\ \vec{0} \end{pmatrix}.$$

The distance r can be rewritten with the scalar product

$$\langle p_0, p_r \rangle_{\mathcal{L}} = \frac{1}{-\kappa}\cosh\big(r\sqrt{-\kappa}\big),$$
$$r = \frac{1}{\sqrt{-\kappa}}\cosh^{-1}(-\kappa\langle p_0, p_r \rangle_{\mathcal{L}}).$$

Exponential map

Let $v = (0, \hat{v})$ and note that $\langle p_0, v \rangle_{\mathcal{L}} = 0$, so v is an element of the tangent space. A more general geodesic at a point p_0 in direction v reads

$$p_r = \exp_{p_0}(r, v) = \cosh\bigl(r\sqrt{-\kappa}\bigr)p_0 + \sinh\bigl(r\sqrt{-\kappa}\bigr)v/\sqrt{-\kappa},$$

and is the intersection of a Minkowski hyperplane with the hyperboloid.

This is the exponential map, which lifts points from the tangent space $\mathrm{T}_{x(0)} \sim \mathbb{R}^n$ to the hyperboloid of dimension n.

Its inverse, logarithmic map, projects points from the hyperboloid to the tangent space.

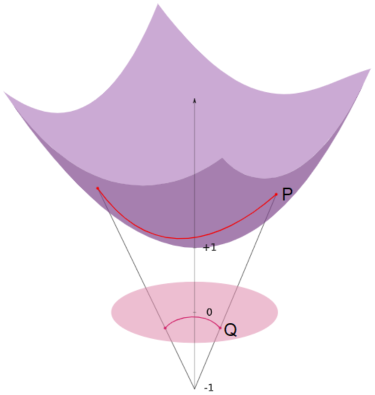
The name comes from combining many infinitesimal movements in the same direction.

From Lorentz hyperboloid to Poincaré disk

The **Poincaré model** is a projection of the Lorentz hyperboloid model to the unit disk/ball via the point $(-1/\sqrt{-\kappa}, \vec{0})$, and vice versa.

$$p_{r,\lambda} = \lambda p_r + (1-\lambda)\begin{pmatrix} -1/\sqrt{-\kappa} \\ \vec{0} \end{pmatrix}$$

$$= \frac{1}{\sqrt{-\kappa}}\begin{pmatrix} \lambda \cosh\left(r\sqrt{-\kappa}\right) - (1-\lambda) \\ \hat{v}\lambda \sinh\left(r\sqrt{-\kappa}\right) \end{pmatrix} \overset{!}{=} \begin{pmatrix} 0 \\ q \end{pmatrix},$$



Poincaré disk: algebra

A parametrization then is

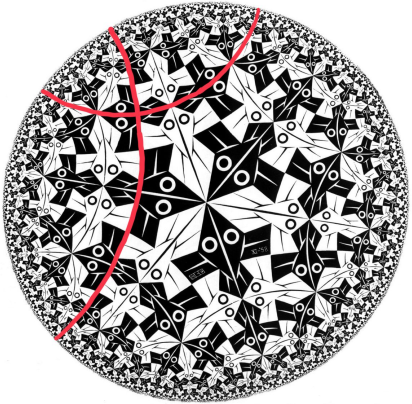
$$q = \frac{\hat{v}}{\sqrt{-\kappa}} \frac{\sinh\bigl(r\sqrt{-\kappa}\bigr)}{\cosh\bigl(r\sqrt{-\kappa}\bigr) - 1} = \frac{\hat{v}}{\sqrt{-\kappa}} \tanh \frac{r\sqrt{-\kappa}}{2},$$

which gives $|q| < \sqrt{-\kappa}$, a disk/ball without shell.

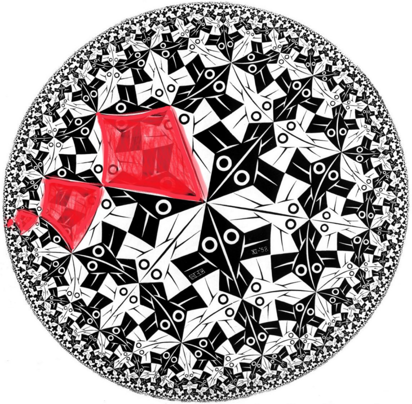
The distance between two points in the Poincaré model is given by

$$d(p, q) = \frac{1}{\sqrt{-\kappa}} \cosh^{-1}\left(1 + \frac{2|p-q|^2}{(1-|p|^2)(1-|q|^2)}\right).$$

Poincaré disk: graphics



Geodesics are arcs of circles that
meet the boundary at right angles.



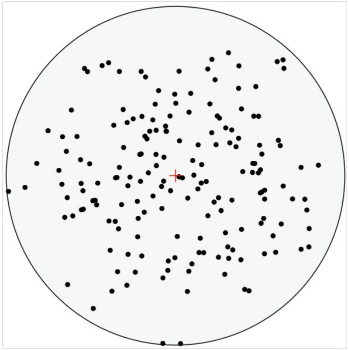
Areas and distances
appear smaller at the boundary.

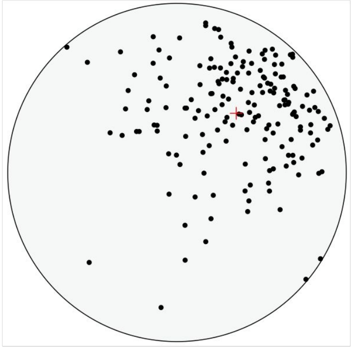
The origin

Is the origin special
in the Lorentz hyperboloid
and in the Poincaré ball?

Hyperbolic isometries

A **hyperbolic translation** τ_x moves 0 to x keeping all pairwise distances constant.
Other names: Lorentz boost, Möbius transformation, gyrovectorspace addition





Formulas for hyperbolic translations

Lorentz hyperboloid (Lorentz boost)

$$\tau_x(y) = \Lambda_x y \quad \text{where} \quad \Lambda_x = \begin{pmatrix} x_0 & \vec{x}^T \\ \vec{x} & \sqrt{\mathbb{I} + \vec{x}\vec{x}^T} \end{pmatrix} \tag{1}$$

Poincaré ball (gyrovectorspace addition)

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

Note $p \oplus q \neq q \oplus p$.

Gyrovectorspace calculus

Gyrovectorspace addition:

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

Gyrovectorspace scalar product:

$$r \otimes p = p \otimes r = \tanh\big(r \tanh^{-1} |p|\big) \frac{p}{|p|}$$

Geodesic arc from p to q:

$$\lambda(t) = p \oplus ([(-p) \oplus q] \otimes t), \quad t \in [0, 1]$$

This is similar to the Euclidean formula $\lambda(t) = p + (q - p)t$.

References I

- Carroll, Sean M. *Lecture notes on general relativity*. 1997. URL: https://arxiv.org/pdf/gr-qc/9712019.
- Loustau, Brice. *Hyperbolic geometry*. 2020. URL: https://brice.loustau.eu/ressources/book.pdf.
- Mettes, Pascal et al. "Hyperbolic deep learning in computer vision: A survey". In: *International Journal of Computer Vision* (2024), pp. 1–25. URL: https://arxiv.org/pdf/2305.06611.
- — .*Hyperbolic Representation Learning for Computer Vision*. URL: https://sites.google.com/view/hyperbolic-tutorial-eccv22.

Universität Basel

HSLU Hochschule Luzern

Hyperbolic Learning in Action

Deep Learning in Hyperbolic Space

Lionetti
Simone

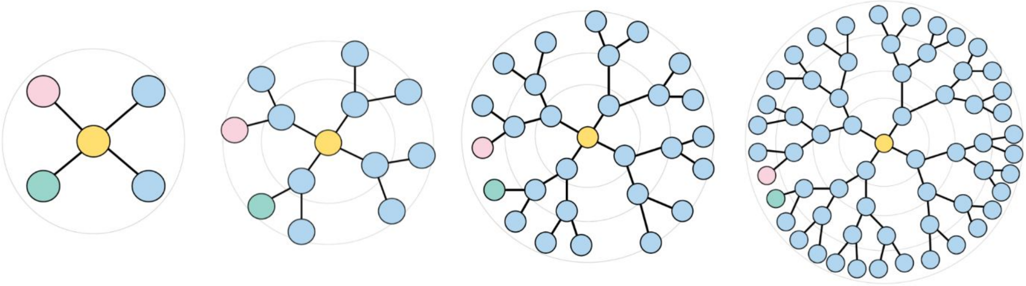
Gonzalez-Jimenez
Alvaro

Friday 14th February, 2025

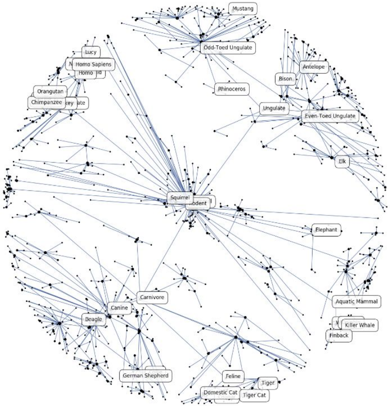
Embedding Hierarchies

The trouble with embedding hierarchies

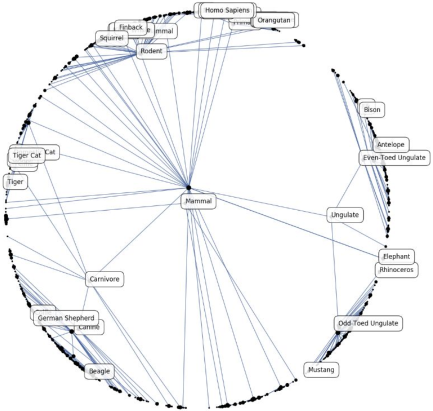
Hierarchies grow exponenitally in depth, Euclidean spaces grows linearly with norm.



Poincaré Embeddings



(a) Intermediate embedding after 20 epochs



(b) Embedding after convergence

Nickel and Kiela, *Poincaré Embeddings for Learning Hierarchical Representations*

Optimizing Poincaré Embeddings

Nodes:
$\mathcal{S} = \{x_i\}_{i=1}^{n}$

Parent-child relations:
$\mathcal{D} = \{(u, v)\}$

Non-Parent-child relations:
$\mathcal{N}(u) = \{v' \,|\, (u, v') \notin \mathcal{D}\} \cup \{v\}$

Optimizing Poincaré Embeddings

Nodes:
$\mathcal{S} = \{x_i\}_{i=1}^{n}$

Parent-child relations:
$\mathcal{D} = \{(u, v)\}$

Non-Parent-child relations:
$\mathcal{N}(u) = \{v' | \, (u, v') \notin \mathcal{D}\} \cup \{v\}$

Hyperbolic representation of nodes: $\Theta = \{\theta_i\}_{i=1}^{n}$

$$\Theta' \leftarrow \operatorname{argmin} \mathcal{L}(\Theta) \quad \text{s.t.} \forall \theta_i \in \Theta : \|\theta_i\| < 1 \tag{1}$$

Pull parent-child nodes, push others.

$$\mathcal{L}(\Theta) = \sum_{(u,v) \in \mathcal{D}} \log \frac{e^{-d(\mathbf{u},\mathbf{v})}}{\sum_{\mathbf{v}' \in \mathcal{N}(u)} e^{-d(\mathbf{u},\mathbf{v}')}} \tag{2}$$

$$d(\mathbf{u}, \mathbf{v}) = \operatorname{arcosh}\left(1 + 2 \frac{\|\mathbf{u} - \mathbf{v}\|^2}{\left(1 - \|\mathbf{u}\|^2\right)\left(1 - \|\mathbf{v}\|^2\right)}\right) \tag{3}$$

Optimizing Poincaré Embeddings

$$\theta_{t+1} = \Re_{\theta_t}\left(-\eta_t \nabla_R \mathcal{L}\left(\theta_t\right)\right) \tag{4}$$

Optimize node embeddings with Riemmanian gradient descent.

$$\theta_{t+1} \leftarrow \text{proj}\left(\theta_t - \eta_t \frac{\left(1 - \|\theta_t\|^2\right)^2}{4} \nabla_E\right) \tag{5}$$

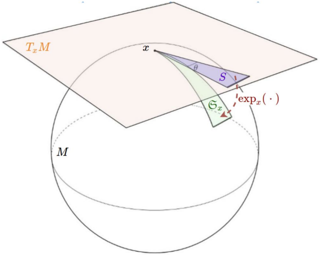
Riemmanian gradient descent = Standard gradient + scaling + projection.

Poincaré Embeddings

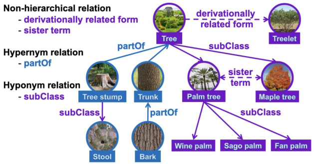
			Dimensionality					
			5	10	20	50	100	200
WordNet Reconstruction	Euclidean	Rank	3542.3	2286.9	1685.9	1281.7	1187.3	1157.3
		MAP	0.024	0.059	0.087	0.140	0.162	0.168
	Translational	Rank	205.9	179.4	95.3	92.8	92.7	91.0
		MAP	0.517	0.503	0.563	0.566	0.562	0.565
	Poincaré	Rank	4.9	4.02	3.84	3.98	3.9	**3.83**
		MAP	0.823	0.851	0.855	0.86	0.857	**0.87**
WordNet Link Pred.	Euclidean	Rank	3311.1	2199.5	952.3	351.4	190.7	81.5
		MAP	0.024	0.059	0.176	0.286	0.428	0.490
	Translational	Rank	65.7	56.6	52.1	47.2	43.2	40.4
		MAP	0.545	0.554	0.554	0.56	0.562	0.559
	Poincaré	Rank	5.7	**4.3**	4.9	4.6	4.6	4.6
		MAP	0.825	0.852	0.861	**0.863**	0.856	0.855

		Dimensionality							
		Reconstruction				Link Prediction			
		10	20	50	100	10	20	50	100
AstroPh N=18,772; E=198,110	Euclidean	0.376	0.788	0.969	0.989	0.508	0.815	0.946	0.960
	Poincaré	0.703	0.897	0.982	0.990	0.671	0.860	0.977	0.988
CondMat N=23,133; E=93,497	Euclidean	0.356	0.860	0.991	0.998	0.308	0.617	0.725	0.736
	Poincaré	0.799	0.963	0.996	0.998	0.539	0.718	0.756	0.758
GrQc N=5,242; E=14,496	Euclidean	0.522	0.931	0.994	0.998	0.438	0.584	0.673	0.683
	Poincaré	0.990	0.999	0.999	0.999	0.660	0.691	0.695	0.697
HepPh N=12,008; E=118,521	Euclidean	0.434	0.742	0.937	0.966	0.642	0.749	0.779	0.783
	Poincaré	0.811	0.960	0.994	0.997	0.683	0.743	0.770	0.774

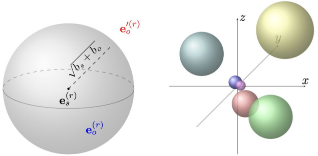
Improving Poincaré Embeddings



Hyperbolic entailment cones.

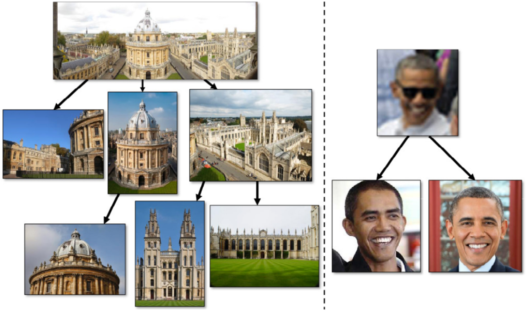


Multi-relational Poincaré embeddings.



Modelling heterogeneous hierarchies.

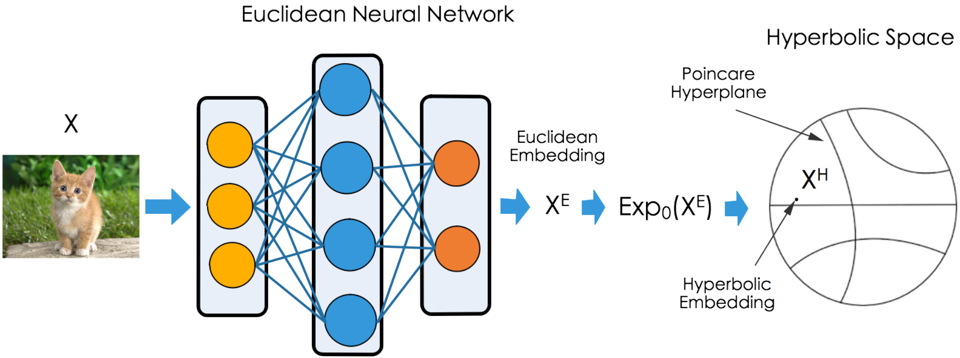
Hyperbolic Image Embeddings



Encoder	Dataset			
	CIFAR10	CIFAR100	CUB	*Mini*ImageNet
Inception v3 [49]	0.25	0.23	0.23	0.21
ResNet34 [14]	0.26	0.25	0.25	0.21
VGG19 [42]	0.23	0.22	0.23	0.17

Khrulkov et al., *Hyperbolic Image Embeddings*

Convolutional Networks with Hyperbolic Embeddings

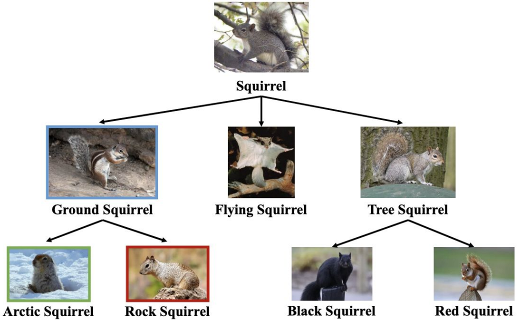


Guo et al., "Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers"

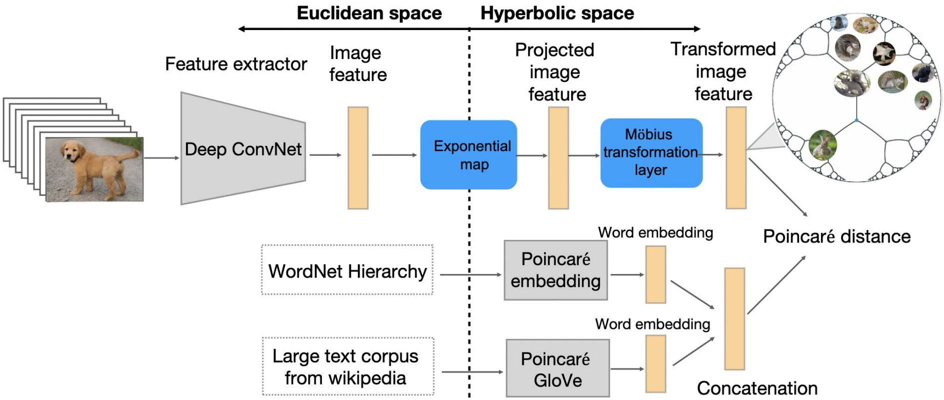
Zero-Shot Generalization

Hyperbolic Zero-Shot Visual Embedding Learning

If labels are hierarchical, does a hyperbolic embedding eneable zero-shot generalization?



Hyperbolic Zero-Shot Visual Embedding Learning



Liu et al., "Hyperbolic Visual Embedding Learning for Zero-Shot Recognition"

Hyperbolic Zero-Shot Visual Embedding Learning

If prior knowledge is a hierarchy then use hyperbolic geometry for zero-shot learning.

Data Set	Model	Hierarchical precision@k(%)				
		1	2	5	10	20
2-hops & Their Parents	DeViSE	3.2	5.3	9.5	15.6	21.2
	DeViSE*	4.5	7.0	9.9	15.6	22.0
	ConSE	4.2	6.8	12.3	18.5	25.1
	GCNZ	9.2	15.6	27.5	36.8	44.5
	Ours	**16.6**	**24.3**	**43.8**	**58.6**	**70.3**
3-hops & Their Parents	DeViSE	1.3	2.1	3.3	4.9	7.3
	DeViSE*	1.7	2.6	4.4	6.6	9.3
	ConSE	1.9	2.6	4.4	7.2	9.7
	GCNZ	2.7	4.6	8.2	12.5	15.1
	Ours	**7.9**	**12.5**	**21.4**	**28.7**	**37.5**
All	DeViSE	0.9	1.5	2.9	4.4	6.5
	DeViSE*	1.0	1.6	2.9	4.4	6.5
	ConSE	1.5	2.4	4.2	6.5	9.7
	GCNZ	2.2	3.8	7.2	10.5	13.9
	Ours	**5.1**	**6.9**	**12.9**	**16.5**	**19.3**

DeViSE: teddy, orangutan, valley, langur, cliff

GCNZ: phalanger, **red squirrel**, kangaroo, lemur, tree wallaby

Ours: **red squirrel**, **tree squirrel***, squirrel, kangaroo, phalanger

DeViSE: rugby ball, soccer ball, golf ball, basketball, cricket

GCNZ: **volleyball**, basketball, golf ball, punching bag, rugby ball

Ours: **volleyball**, **ball***, basketball, rugby ball, soccer ball

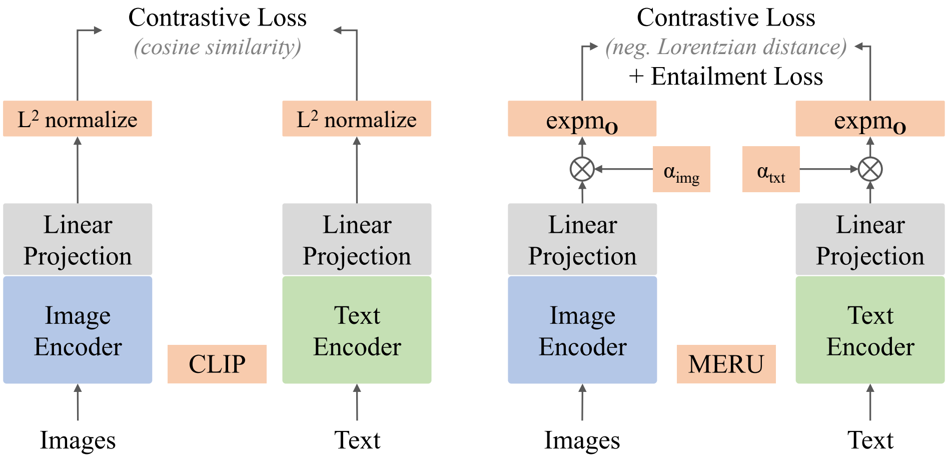
DeViSE: bullet train, freight car, school bus, police van, minibus

GCNZ: mail train, express, **passenger train**, cargo ship, shuttle bus

Ours: **passenger train**, **railroad train***, bus, school bus, trolleybus

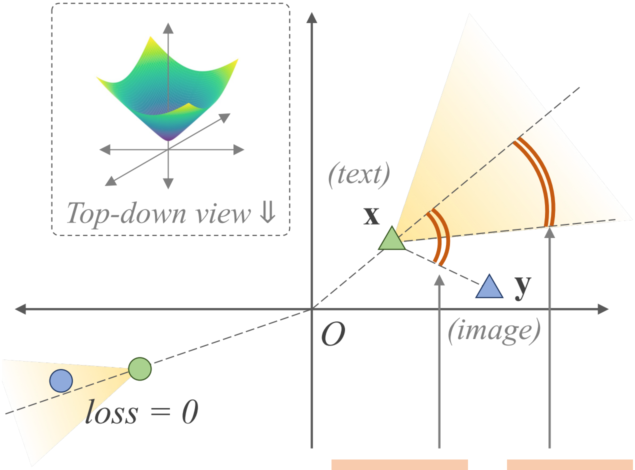
Liu et al., "Hyperbolic Visual Embedding Learning for Zero-Shot Recognition"

Hyperbolic Image-Text Representations



Desai et al., "Hyperbolic Image-text Representations"

Entailment Cones



$$loss = \text{ext}(\mathbf{x}, \mathbf{y}) - \text{aper}(\mathbf{x})$$

Desai et al., "Hyperbolic Image-text Representations"

MERU

MERU	CLIP
avocado toast	*avocado toast*
healthy breakfast	*delicious*
delicious	↓
homemade	↓
fresh	↓
[ROOT]	[ROOT]

MERU	CLIP
brooklyn bridge	*photo of brooklyn bridge, new york*
new york city	*new york city*
city	*new york*
outdoors	↓
day	↓
[ROOT]	[ROOT]

MERU	CLIP
taj mahal	*taj mahal through an arch*
monument	*travel*
architecture	*inspiration*
travel	↓
day	↓
[ROOT]	[ROOT]

MERU	CLIP
sydney opera house	*sydney opera house*
opera house	*opera house*
holiday	*gift*
day	*beauty*
[ROOT]	[ROOT]

Desai et al., "Hyperbolic Image-text Representations"

MERU

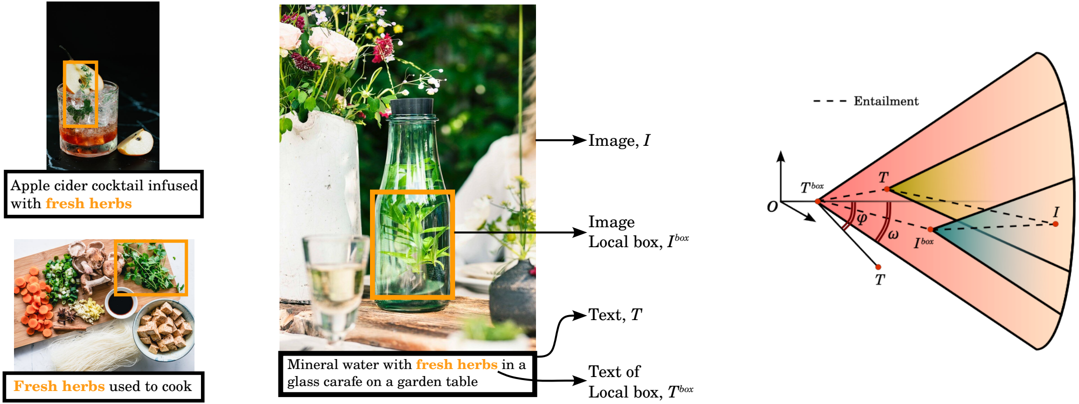
Table 2. **Zero-shot image classification.** We train MERU and CLIP models with varying parameter counts and transfer them *zero-shot* to 20 image classification datasets. Best performance in every column is highlighted in green. Hyperbolic representations from MERU match or outperform CLIP on 13 out of the first 16 datasets. On the last four datasets (gray columns), both MERU and CLIP have *near-random* performance, as concepts in these datasets are not adequately covered in the training data.

		ImageNet	Food-101	CIFAR-10	CIFAR-100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	STL-10	EuroSAT	RESISC45	Country211	MNIST	CLEVR	PCAM	SST2
ViT S/16	CLIP	34.3	74.5	**60.1**	24.4	**33.8**	27.5	**11.3**	**1.4**	15.0	**73.7**	63.9	47.0	88.2	18.6	31.4	**5.2**	10.0	19.4	50.2	50.1
	MERU	**34.4**	**75.6**	52.0	**24.7**	33.7	**28.0**	11.1	1.3	**16.2**	72.3	**64.1**	**49.2**	**91.1**	**30.4**	**32.0**	4.8	7.5	14.5	51.0	50.0
ViT B/16	CLIP	**37.9**	**78.9**	65.5	**33.4**	33.3	29.8	**14.4**	1.4	17.0	77.9	**68.5**	50.9	92.2	25.6	31.0	**5.8**	10.4	14.3	54.1	51.5
	MERU	37.5	78.8	**67.7**	32.7	**34.8**	**30.9**	14.0	**1.7**	**17.2**	**79.3**	**68.5**	**52.1**	**92.5**	**30.2**	**34.5**	5.6	13.0	13.5	49.8	49.9
ViT L/16	CLIP	38.4	80.3	**72.0**	**36.4**	36.3	32.0	**18.0**	1.1	16.5	78.8	**68.3**	48.6	**93.7**	26.7	35.4	6.1	14.8	13.6	51.2	51.1
	MERU	**38.8**	**80.6**	68.7	35.5	**37.2**	**33.0**	16.6	**2.2**	**17.2**	**80.0**	67.5	**52.1**	**93.7**	**28.1**	**36.5**	**6.2**	11.8	13.1	52.7	49.3

Desai et al., "Hyperbolic Image-text Representations"

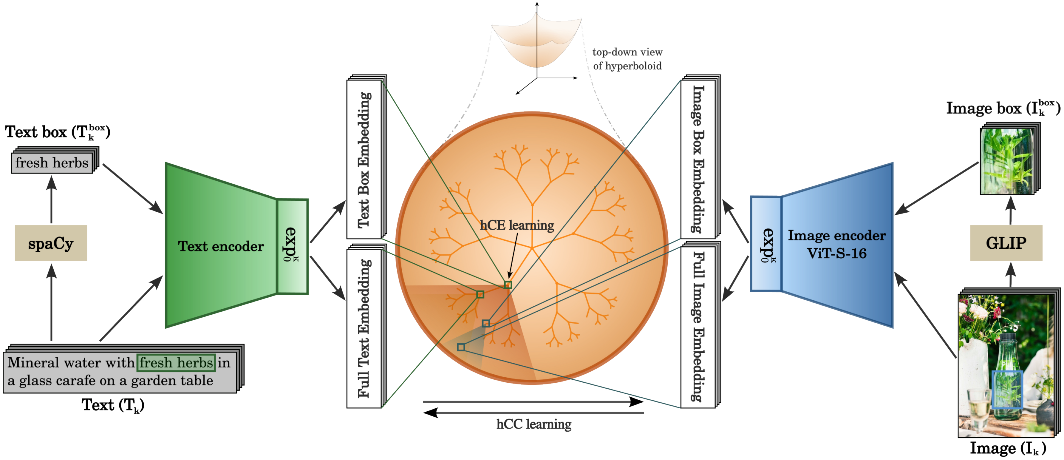
Compositional Entailment Learning

An image is not only described by a sentence but is itself a composition of multiple object boxes, each with their own textual description



Pal et al., "Compositional Entailment Learning for Hyperbolic Vision-Language Models"

Compositional Entailment Learning



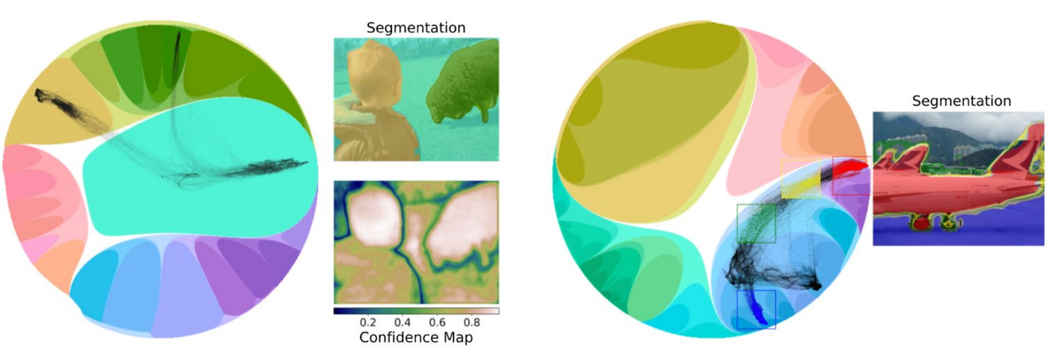
Pal et al., "Compositional Entailment Learning for Hyperbolic Vision-Language Models"

Compositional Entailment Learning

Pal et al., "Compositional Entailment Learning for Hyperbolic Vision-Language Models"

Robustness and Uncertainty

Hyperbolic Segmentation

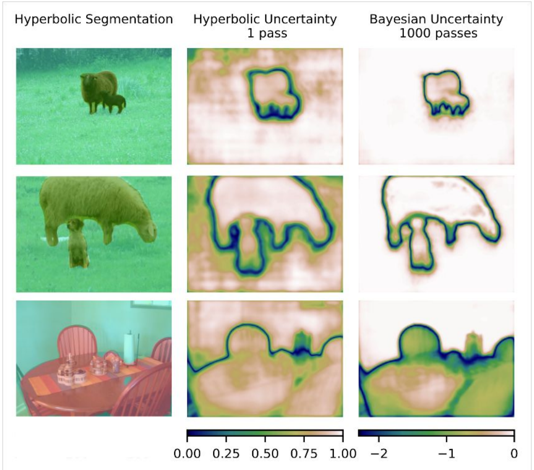


(a) Prediction uncertainty for free

(b) Boundary information for free

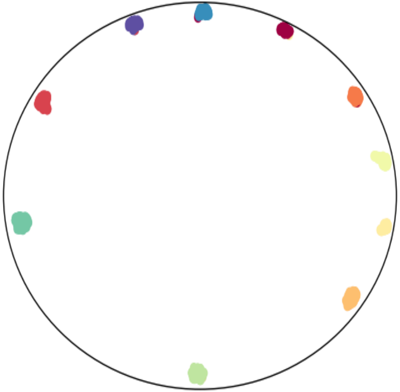
GhadimiAtigh et al., *Hyperbolic Image Segmentation*

Uncertainty and boundary information for free



GhadimiAtigh et al., *Hyperbolic Image Segmentation*

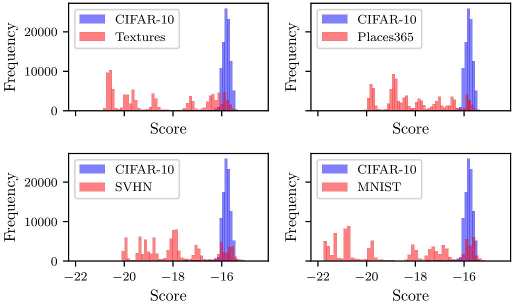
How to exploit robustness properties?

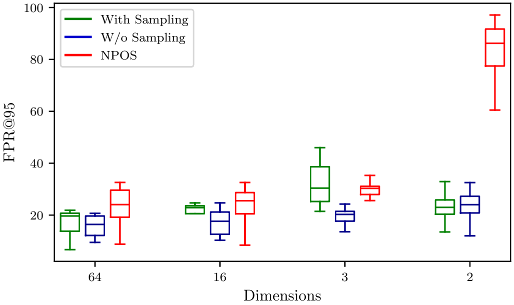


- Learn in-distribution representations promoting low variations and high separation.
- Hyperbolic geometry offers *more space* than Euclidean!

Gonzalez-Jimenez et al., *Hyperbolic Metric Learning for Visual Outlier Detection*

Hyperbolic Outlier Detection



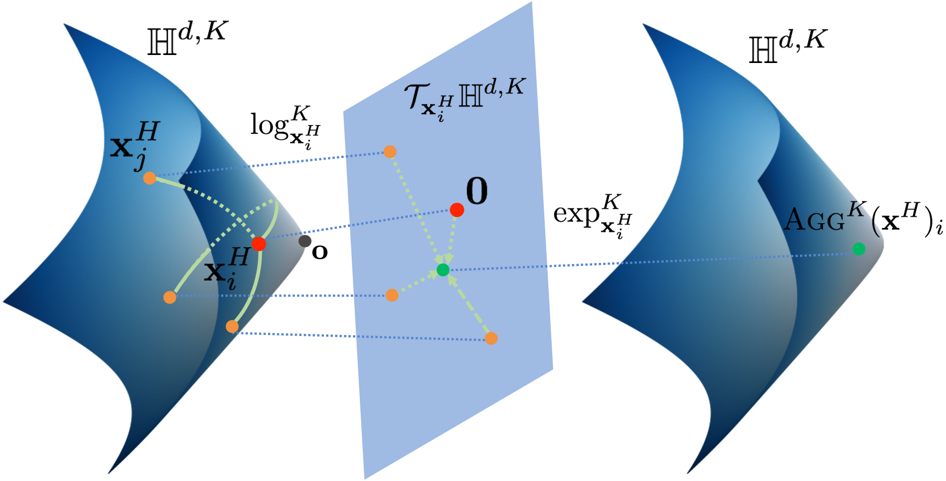


Gonzalez-Jimenez et al., *Hyperbolic Metric Learning for Visual Outlier Detection*

Fully Hyperbolic Networks

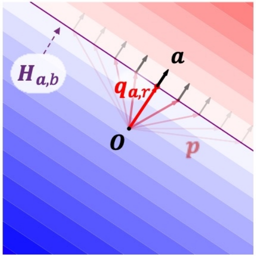
Extend Hyperbolic Space for the entire network

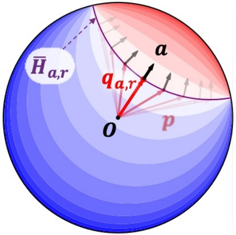
Mapping back and forth between hyperbolic and Euclidean manifolds.



Move Everything to Hyperbolic Space

The current methods depend on the tangent space for several operations and the frequent back and forth mapping is both expensive and prone to a loss of data.





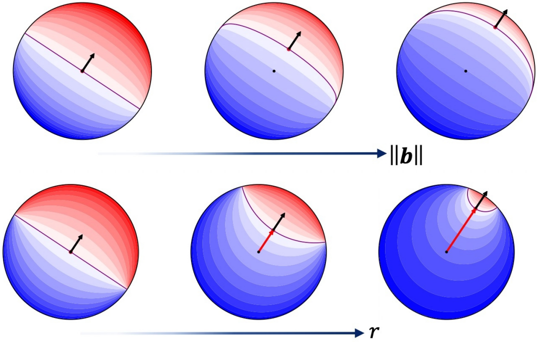
Hyperbolic Network ++

Poincaré Ball

$$y = \exp_0^c\left(W \log_0^c(x)\right) \oplus_c b \qquad (6)$$

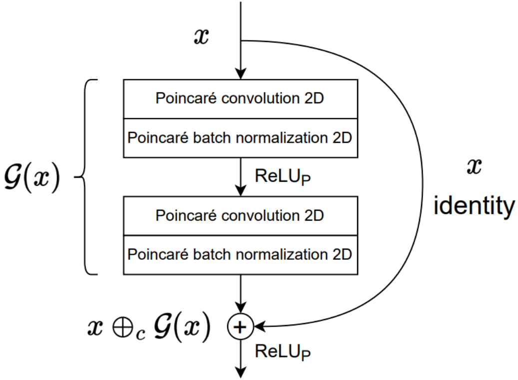
Hyperbolic Neural Networks++

$$y = \mathcal{F}^c(x; Z, r) = \mathrm{w}\left(1 + \sqrt{1 + c\|w\|^2}\right)^{-1} \qquad (7)$$



Shimizu, Mukuta, and Harada, *Hyperbolic Neural Networks++*

Poincare ResNet

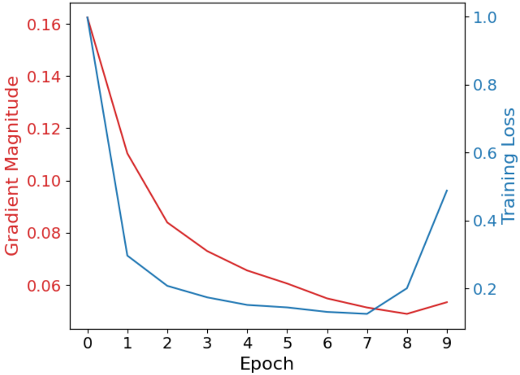


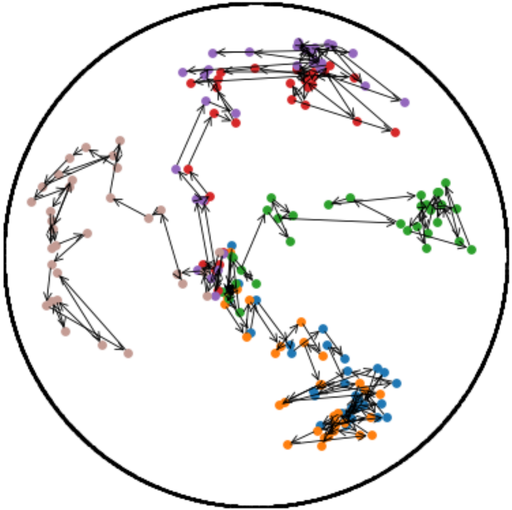
- Extend Linear Layer from Hyperbolic Neural Network++ for Convolutions.
- Poincare midpoint batch normalization for faster and equally effective alternative to Frechet Mean.
- Poincare Resnets are (i) more robust to out-of-distribution samples, (ii) more robust to adversarial attacks and (iii) complementary to Euclidean networks.

Spengler, Berkhout, and Mettes, *Poincaré ResNet*

Weaknesses

Gradients Vanishing





Guo et al., "Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers"

Numerical Instability

- It will sometimes lead to catastrophic NaN problems, encountering unrepresentable values in floating point arithmetic.
- Under the 64 bit arithmetic system, the Poincare ball has a relatively larger capacity than the Lorentz model for correctly representing points.
- Lorentz model is superior to the Poincare ball from the perspective of optimization.

Mishne et al., "The Numerical Stability of Hyperbolic Representation Learning"

References I

- Desai, Karan et al. “Hyperbolic Image-text Representations”. In: *Proceedings of the 40th International Conference on Machine Learning*. PMLR, July 2023, pp. 7694–7731. (Visited on 02/05/2025).
- GhadimiAtigh, Mina et al. *Hyperbolic Image Segmentation*. Mar. 2022. DOI: 10.48550/arXiv.2203.05898. (Visited on 02/05/2025).
- Gonzalez-Jimenez, Alvaro et al. *Hyperbolic Metric Learning for Visual Outlier Detection*. Sept. 2024. DOI: 10.48550/arXiv.2403.15260. (Visited on 02/05/2025).
- Guo, Yunhui et al. “Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers”. In: *Computer Vision and Pattern Recognition* (2021). DOI: 10.1109/CVPR52688.2022.00010.
- Khrulkov, Valentin et al. *Hyperbolic Image Embeddings*. Mar. 2020. DOI: 10.48550/arXiv.1904.02239. (Visited on 02/05/2025).

References II

Liu, Shaoteng et al. "Hyperbolic Visual Embedding Learning for Zero-Shot Recognition". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020, pp. 9270–9278. DOI: 10.1109/CVPR42600.2020.00929. (Visited on 02/11/2025).

Mishne, Gal et al. "The Numerical Stability of Hyperbolic Representation Learning". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23-29 Jul 2023, pp. 24925–24949. URL: https://proceedings.mlr.press/v202/mishne23a.html.

Nickel, Maximilian and Douwe Kiela. *Poincaré Embeddings for Learning Hierarchical Representations*. May 2017. DOI: 10.48550/arXiv.1705.08039. (Visited on 02/26/2024).

References III

- Pal, Avik et al. "Compositional Entailment Learning for Hyperbolic Vision-Language Models". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=3i13Gev2hV.
- Shimizu, Ryohei, Yusuke Mukuta, and Tatsuya Harada. *Hyperbolic Neural Networks++*. Mar. 2021. DOI: 10.48550/arXiv.2006.08210. (Visited on 11/20/2024).
- Spengler, Max van, Erwin Berkhout, and Pascal Mettes. *Poincaré ResNet*. Dec. 2023. DOI: 10.48550/arXiv.2303.14027. (Visited on 02/05/2025).

Hyperbolic Learning in Action

Tools for Hyperbolic Learning

Lionetti
Simone

Gonzalez-Jimenez
Alvaro

Friday 14th February, 2025

Geomstats

- An extensive library for differential geometry, supporting a wide range of manifolds and operations.
- Offers thorough documentation and tutorials, which help users get started quickly.
- Well-maintained and actively developed with frequent updates.
- Lacks in-depth focus on hyperbolic space, leading to missing important features and models.

Geomstats

Geoopt

- Started as *just for fun* paper implementation and grow to a Python Package. [1]
- Support various hyperbolic models and operations.
- The most widely used library in literature for manifold-based optimization.
- Lacks active maintenance, with outdated implementations for key operations such as sinh, cosh, etc.
- Performance issues and steep learning curve for beginners.

[1] https://www.youtube.com/watch?v=6VZ0Gk4QMME

HypLL

- A recent library with a strong focus on hyperbolic space.
- Provides support for hyperbolic layers and operations, designed like PyTorch (`hypll.nn`, `hypll.optim`).
- User-friendly for creating fully hyperbolic networks.
- Only support Poincaré Ball, other models will be implemented.

Hypll Overview

```python
from hypll.tensors import TangentTensor
from hypll.optim import RiemannianAdam
from hypll.manifolds.poincare_ball import Curvature, PoincareBall
from models import hyperbolic_model
...

manifold = PoincareBall(c=Curvature(value=0.1, requires_grad=True))
model = hyperbolic_model(manifold=manifold)

optimizer = RiemannianAdam(model.parameters(), lr=0.001)
criterion = nn.CrossEntropyLoss()

for epoch in range(100):
    running_loss = 0.0
    for i, data in enumerate(trainloader, 0):
        inputs, labels = data[0].to(device), data[1].to(device)

        tangents = TangentTensor(data=inputs, man_dim=1, manifold=manifold)
        manifold_inputs = manifold.expmap(tangents)

        optimizer.zero_grad()
        outputs = model(manifold_inputs)
        loss = criterion(outputs.tensor, labels)
        loss.backward()
        optimizer.step()
        ...
```

Library	Advantages	Disadvantages
Geomstats	Extensive support for differential geometry Well maintained and documented	Not focus for hyperbolic learning Missing operations and features
Geoopt	Support many models (Lorentz, Hyperboloid, Klein, etc.) Rich in hyperbolic operations Widely used in hyperbolic papers	Slow performance (outdated code) Not maintained Difficult for beginners
Hypll	Follows PyTorch style Support hyperbolic layers for Fully Hyperbolic Networks User-friendly	Only Poincare model is supported

Practical Session

The code is at

`https://github.com/Digital-Dermatology/hyperbolic-learning-tutorial-code`

Universität Basel

HSLU Hochschule Luzern

Hyperbolic Learning in Action

Conclusions & Learning

Lionetti
Simone

Gonzalez-Jimenez
Alvaro

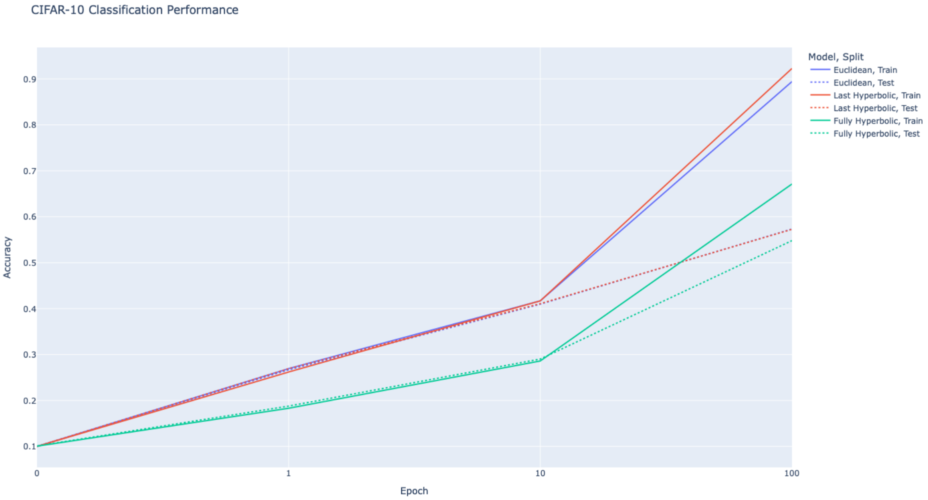
Friday 14th February, 2025

Discussion

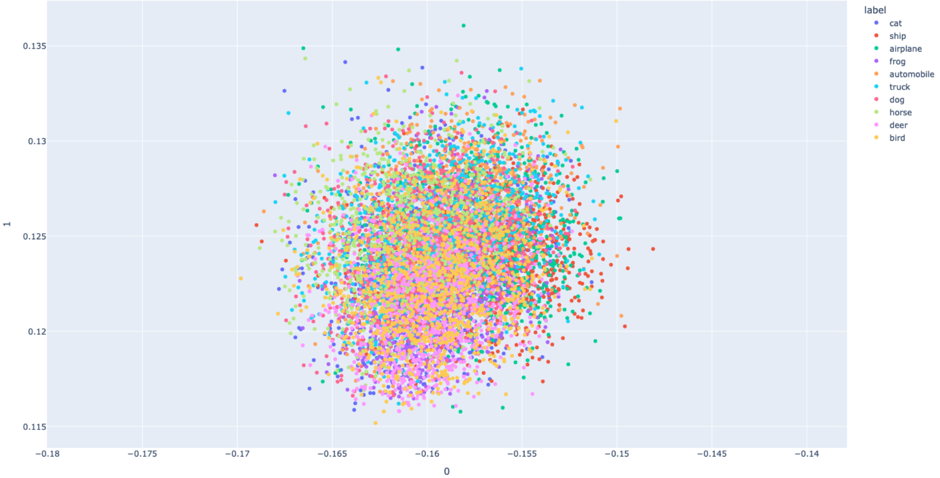
Learnings from the practical session

Tell us what you have discovered or learned from the practical session!

Performance comparison

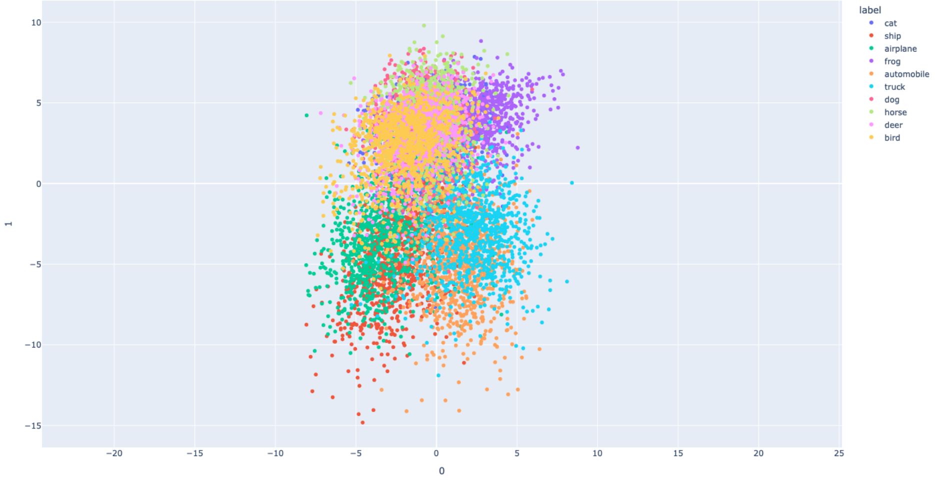


Euclidean, epoch 0



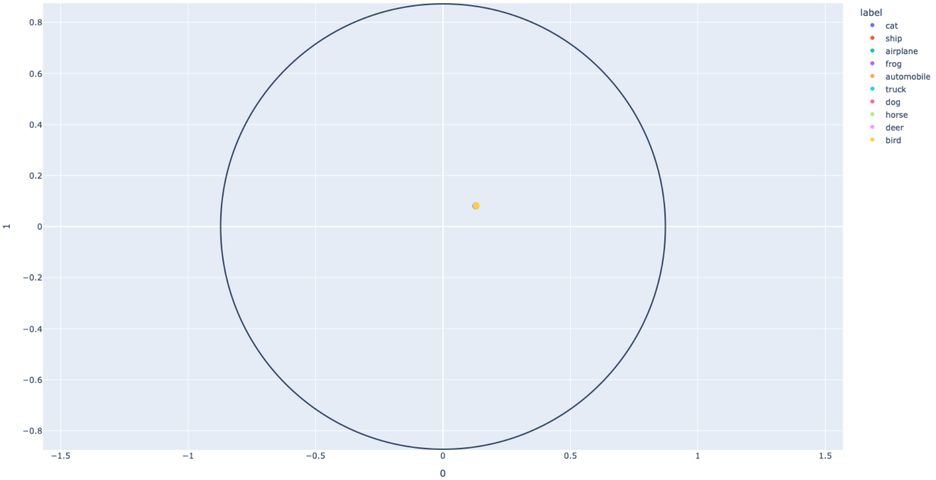
Euclidean, epoch 1

Euclidean, epoch 10

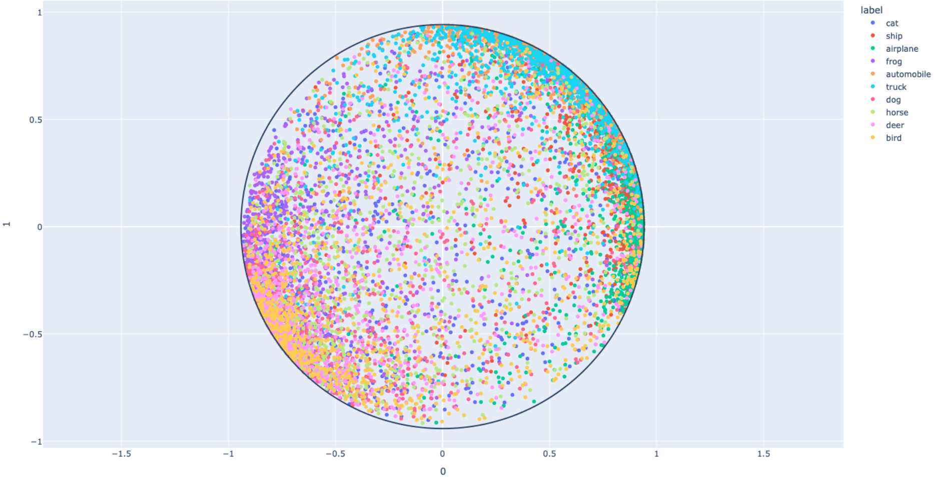


Euclidean, epoch 100

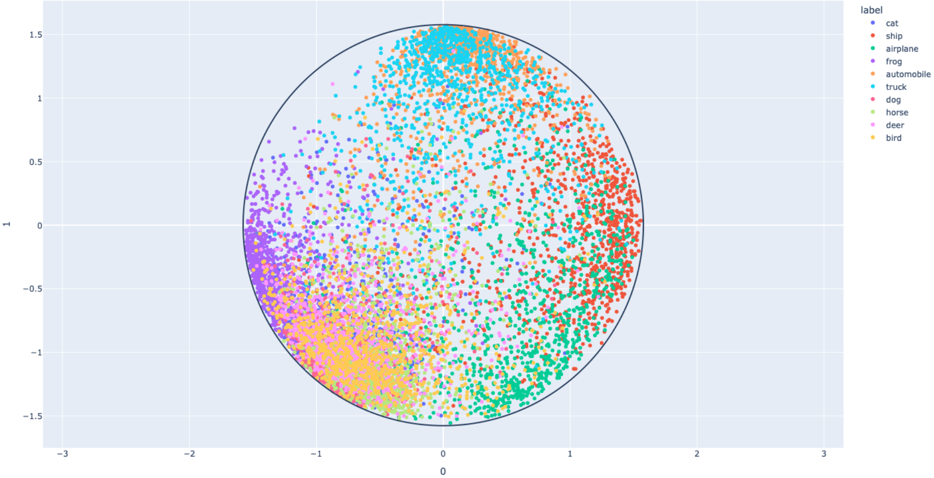
Last hyperbolic, epoch 0



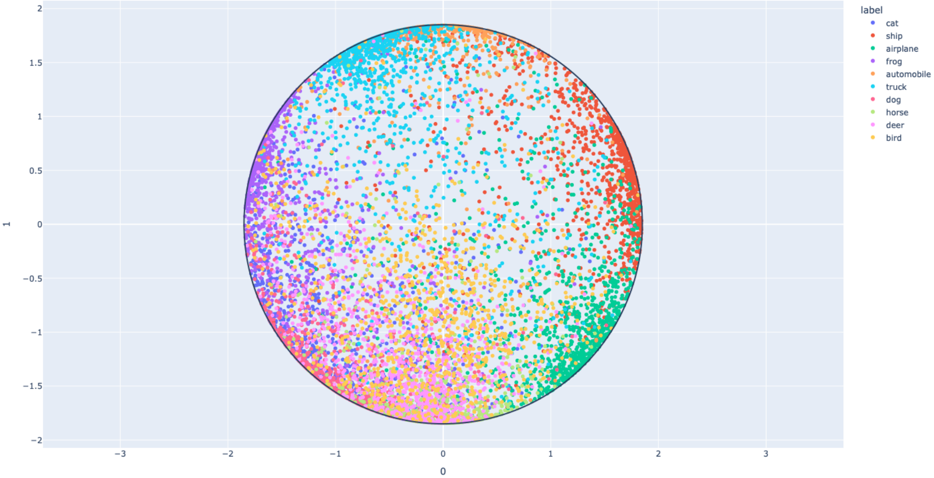
Last hyperbolic, epoch 1



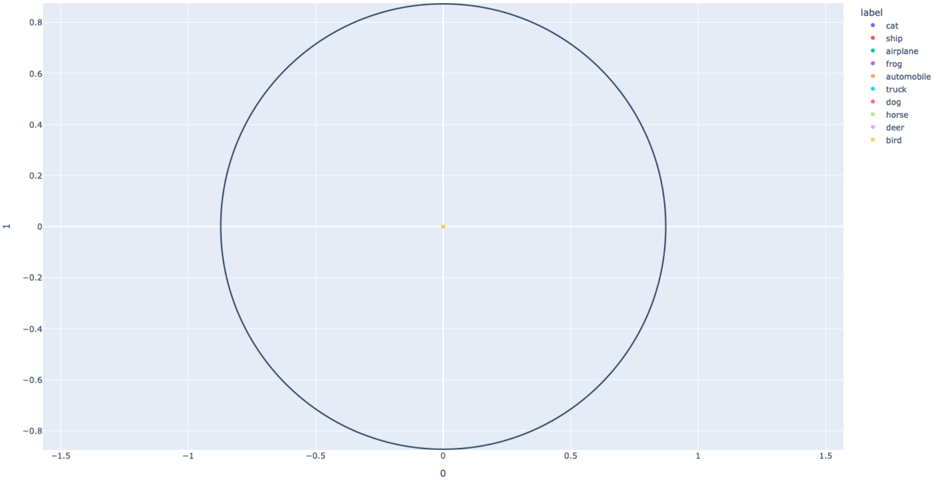
Last hyperbolic, epoch 10



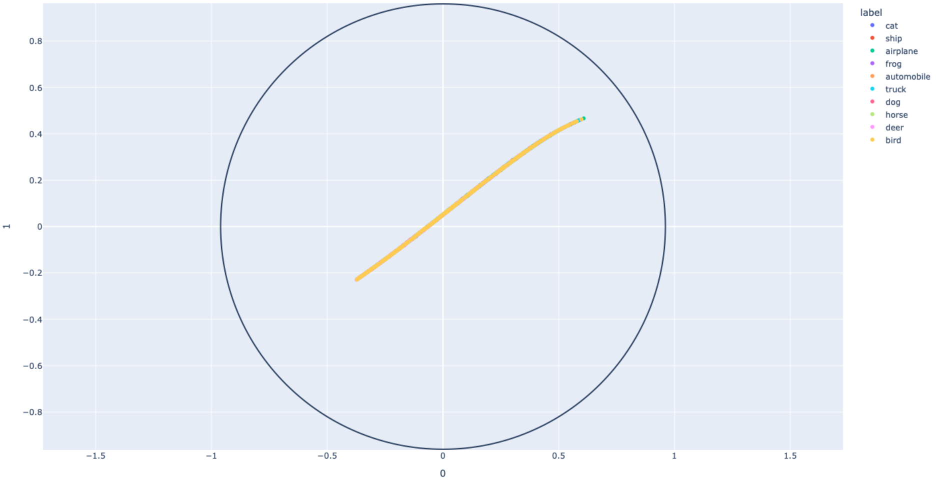
Last hyperbolic, epoch 100



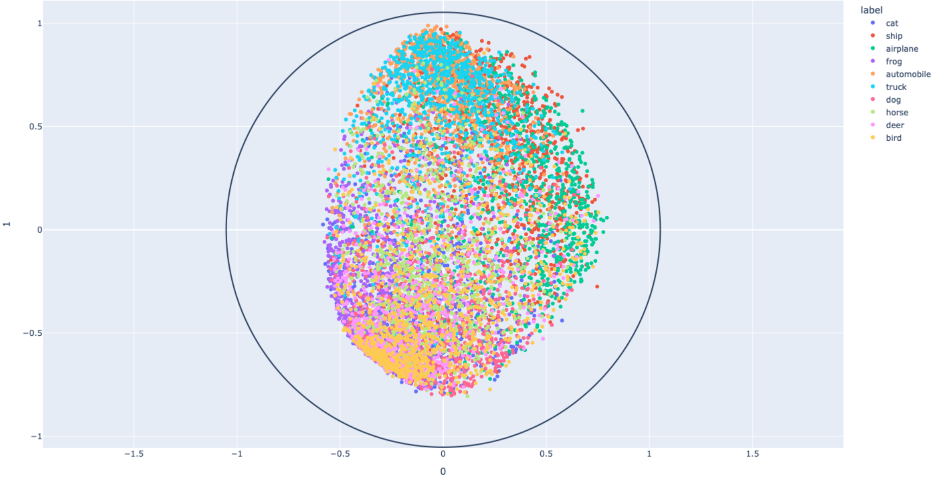
Fully hyperbolic, epoch 0



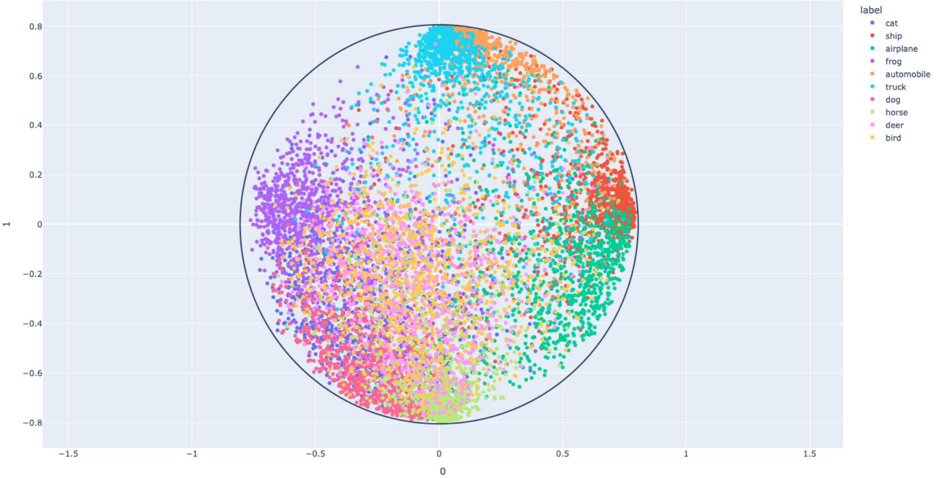
Fully hyperbolic, epoch 1



Fully hyperbolic, epoch 10

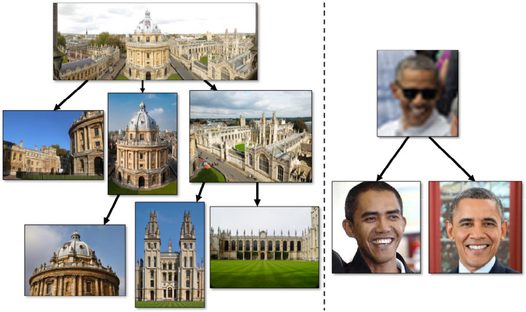


Fully hyperbolic, epoch 100

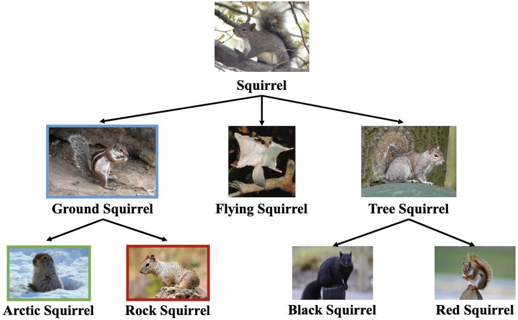


Recap

Why should we care about Hyperbolic Learning

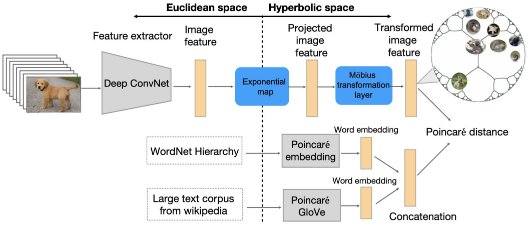


Visual Hierarchies

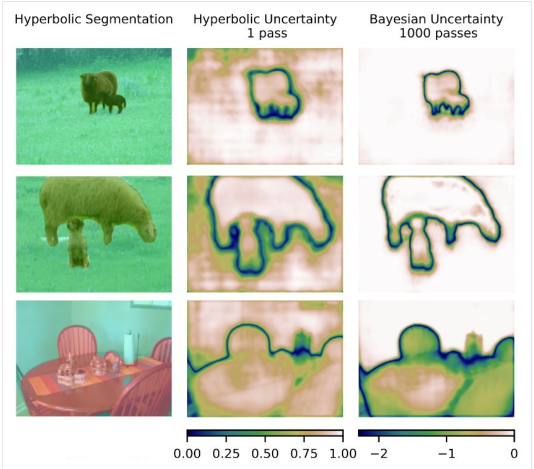


Semantic Hierarchies

Why should we care about Hyperbolic Learning



Zero-Shot Learning



Robustness

Future Potential

- Fully hyperbolic CNNs, Transformers, etc.
- Stable learning on any and all hyperbolic models.
- Fast forward and backward computation.
- Adjust curvature to data and problem.
- What model is suitable for data and problem?
- Large-scale hyperbolic learning.

Thank you

https://forms.gle/KdhQPt6e9NwKUkfGA